Advanced Angular TypeScript Syllabus -2025

Section 1:

Angular essentials Introduction - components - templates , services and more

1. A new starting project and analysing the project structure.
2. Understanding Components & How Content Ends Up On The Screen
3. Creating a First Custom Component.
4. Configuring the Custom Component
5. Configuring the Custom Component
6. Using the Custom Component
7. Styling the Header Component & Adding An Image
8. Managing & Creating Components with the Angular CLI
9. Styling & Using Our Next Custom Component
10. Preparing User Data (To Output Dynamic Content)
11. Storing Data in a Component Class
12. Outputting Dynamic Content with String Interpolation
13. Property Binding & Outputting Computed Values
14. Attribute Binding
15. Using Getters For Computed Values
16. Listening to Events with Event Binding
17. Managing State & Changing Data
18. A Look Behind The Scenes Of Angular's Change Detection Mechanism
19. Introducing Signals
20. We Need More Flexible Components!
21. Defining Component Inputs
22. Required & Optional Inputs
23. We Need Custom Events!
24. Working with Outputs & Emitting Data
25. Using the output() Function
26. Adding Extra Type Information To EventEmitter

Exercise: Create a Configurable Component

27. TypeScript: Working With Potentially Undefined Values & Union Types
28. Accepting Objects As Inputs & Adding Appropriate Typings
29. TypeScript: Type Aliases & Interfaces
30. Outputting List Content
31. Outputting Conditional Content

32. Legacy Angular: Using ngFor & ngIf
33. Adding More Components to the Demo App
34. Outputting User-specific Tasks
35. Outputting Task Data in the Task Component
36. Storing Data Models in Separate Files
37. Dynamic CSS Styling with Class Bindings
38. More Component Communication: Deleting Tasks
39. Creating & Conditionally Rendering Another Component
40. Using Directives & Two-Way-Binding
41. Signals & Two-Way-Binding
42. Handling Form Submission
43. Using the Submitted Data
44. Content Projection with ng-content
45. Transforming Template Data with Pipes
46. Getting Started with Services
47. Getting Started with Dependency Injection
48. More Service Usage & Alternative Dependency Injection Mechanism
49. Using localStorage for Data Storage

Section 2:

Angular essentials - working with Modules.

1. Module Introduction
2. A First Introduction To Angular Modules (NgModule)
3. Creating a First Empty Module
4. Bootstrapping Apps with Angular Modules
5. Declaring & Using Components
6. A First Summary
7. 70. Migrating All Components To Use Modules
8. Creating & Using Shared Modules
9. Creating More Complex Module-based App Structures

Advanced Angular TypeScript Syllabus -2025

Section 3:

Angular essentials - Debugging Angular Apps

1. Understanding Error Messages & fixing Errors
2. Debugging Logical Errors with the Browser DevTools & BreakPoints
3. Exploring the Angular DevTools

Section 4:

Enhancing Elements with Directives (Lite)

1. Understanding Directives
2. The Starting Project
3. Analysing a Built-in Attribute Directive: ngModel
4. Analysing a Built-in Structural Directive: ngIf
5. Getting Started with Custom Directives
6. Using Attribute Directives To Change Element Behavior
7. Working with Inputs in Custom Directives
8. Directives & Dependency Injection
9. Building Another Directive
10. Building a Custom Structural Directive
11. Structural Directives & Syntactic Sugar
12. Host Directives & Composition

Section 5:

Transforming Values with Pipes (Lite)

1. Making Sense of Pipes
2. Using Built-in Pipes
3. More Built-in Pipes Examples
4. Building a First Custom Pipe
5. Using Custom Pipes to Perform Custom Transformations
6. Accepting Parameters in Custom Pipes

7. Chaining Pipes & Being Aware of Limitations
8. Building a Pipe That Sorts Items
9. Understanding How Pipes Are Executed
10. Pure & Impure Pipes
11. Pipe Limitations & When Not To Use Them

Section 6:

Understanding Services & Dependency Injection (Advanced)

1. Creating a Service
2. How NOT To Provide A Service
3. Using Angular's Dependency Injection Mechanism
4. Using The Alternative Dependency Injection Syntax
5. Outsourcing & Reusing Logic with Services
6. Angular Has Multiple Injectors!
7. There Are Multiple Ways Of Providing a Service
8. Providing Services via the Element Injector
9. Understanding the Element Injector's Behaviour
10. Injecting Services Into Services
11. Analysing Dependency Injection with the Angular DevTools
12. 185. Using Custom DI Tokens & Providers
13. Preparing A Non-Class Value For Injection
14. Injecting Other Values (NOT Services)
15. Angular Modules (NgModule) & Dependency Injection
16. Working with Services Without Using Signals

Section 7:

Sending HTTP Requests & Handling Responses (Lite)

1. The Starting Projects: Frontend & Backend
2. How To Connect Angular Apps To A Backend
3. Getting Started with Angular's Http Client

4. Providing the HttpClient when using NgModules
5. Sending a GET Request To Fetch Data
6. Configuring Http Requests
7. Transforming & Using Response Data
8. Showing a Loading Fallback
9. Handling HTTP Errors
10. Sending Data To A Backend
11. More Data Fetching & Some Code Duplication
12. Outsourcing HTTP Request Logic Into A Service
13. Managing HTTP-loaded Data via a Service
14. Implementing Optimistic Updating
15. Potential Problems Introduced by Optimistic Updating
16. Improved Optimistic Updating
17. Implementing App-wide Error Management
18. Practice: Sending DELETE Requests
19. Introducing HTTP Interceptors
20. Optional: Class-based Interceptors
21. Introducing HTTP Response Interceptors

Section 8:

Handling user Input & working with forms (Template -driven & Reactive) (Advanced)

1. Template-driven vs Reactive Forms
2. Template-driven: Registering Form Controls
3. Getting Access to the Angular-managed Form
4. Extracting User Input Values
5. Validating Input with Form Validation Directives
6. Using the Form Validation Status To Provide User Feedback
7. Adding Validation Styles
8. Interacting With The Underlying Form Object In The Component
9. Updating Form Values Programmatically
10. Reactive Forms: Getting Started
11. Syncing Reactive Form Definition & Template

12. Handling Form Submission (Reactive Forms)
13. Adding Validators To Reactive Forms
14. Building Custom Validators
15. Creating & Using Async Validators
16. Interacting with the Form Programmatically
17. Lecture incomplete. Progress cannot be changed for this item.
18. Connecting & Registering Inputs For A Complex Form
19. Working with Nested Form Groups
20. Working with Form Arrays
21. Practice: Adding More Validation
22. Creating Multi-Input Validators / Form Group Validators

Section 9

Routing & Building multi-page Single page applications (Lite)

1. What Is Routing?
2. Enabling Routing & Adding a First Route
3. Rendering Routes
4. Registering Multiple Routes
5. Adding Links The Right Way
6. Styling Active Navigation Links
7. Setting Up & Navigating To Dynamic Routes
8. Extracting Dynamic Route Parameters via Inputs
9. Extracting Dynamic Route Parameters via @Input()
10. Extracting Dynamic Route Parameters via Observables
11. Working with Nested Routes
12. Route Links & Relative Links
13. Accessing Parent Route Data From Inside Nested Routes
14. Loading Data Based On Route Parameters In Child Routes
15. Link Shortcuts & Programmatic Navigation
16. Adding A "Not Found" Route
17. Redirecting Users
18. Splitting Route Definitions Across Multiple Files
19. Activated Route vs Activated Route Snapshot

20. Setting Query Parameters
21. Extracting Query Parameters via Inputs
22. Extracting Query Parameters via Observables
23. Using Query Parameters For Data Manipulation
24. Adding Static Data To Routes
25. Resolving Route-related Dynamic Data
26. Optional: Class-based Resolvers
27. Accessing Route Data In Components
28. Controlling Route Resolver Execution
29. Setting & Resolving Titles
30. Introducing Route Guards
31. Making Sense of The CanDeactivate Guard
32. Improving The CanDeactivate Logic
33. Reloading Pages via the Angular Router & Configuring Programmatic Navigation

Section 10:

Code Splitting & Deferrable Views (Advanced)

1. What Is Lazy Loading / Code Splitting?
2. Introducing Route-based Lazy Loading
3. Implementing Route-based Lazy Loading
4. Lazy Loading Entire Route Groups
5. Using Lazy Loading & Routing to Lazy-load Services
6. Introducing Deferrable Views
7. Defer Loading Until Viewport Visibility
8. Deferrable Views: Using Other Triggers
9. Prefetching Lazy-loaded Code
10. Deferrable Views: Summary

Section 11:

Deploying Angular Apps - CSR, SSR, SGA

(more Advanced from Devops)

1. The Starting Project
2. Preparing a Project For Deployment: Building It For Production
3. Building SPAs: Pros & Cons
4. SPAs: Deployment Example
5. Using "ng add", "ng deploy" & Angular's Built-in Deployment Support
6. Server-side Rendering (SSR) Introduction
7. Setting Up SSR For An Angular App
8. Building and Service an SSR App
9. Authoring SSR-ready Code (Beware of Pitfalls!)
10. SSR and Client-Server Mismatches
11. Static Site Generation (SSG) Introduction
12. Configuring & Using SSG
13. Deployment Methods - A Summary
14. SSR & SSG Deployment Example

NOTE ! !

Syllabus Map

Essentials - All the sub headings will be covered with a full stack package.

Lite - Basics under these headings will be covered in a full stack package .

Advanced & more - Students who join Advance-Angular-TS course will have access to all the topic mentioned above (To enrol to AAT) please contact your Co-ordinator.