**Questions & Assignments List**

**1. First Component with Architecture**

* 1. What is a component based architecture in Angular?

a software **design approach** where the application is built by **composing reusable, self-contained components** that **encapsulate** both the **presentation (UI)** and the **behavior (logic)** of a specific part of the user interface.

* 1. What files does any basic Angular component have?

Component Class ( logic part )

Component Template ( html )

Component Styles

Component Unit Test (Spec file)

Module File (optional)

If the component is part of a feature module

* 1. Differentiate between index.html and component.html?

Index.html --> the **entry point** for the application and is **loaded initially** by the browser.contains a single **<app-root>** tag where the Angular application is **bootstrapped**.

component.html is a **template** file specific to an Angular component.

commonly **including data binding, directives, event handling**, and other Angular-specific features.

1. **Displaying Message Dynamically (Interpolation)**

Is a technique used in Angular to **dynamically display data** in the template of a component using interpolation syntax ({{ }}). inside **two curly brackets.**

**e**g.: {{num1}}, {{ num1 + num2 }}, {{ getName() }}

2.1 Assignment - Displaying sum of two numbers

**3. Understanding @NgModule**

The @NgModule **decorator** is a key feature in Angular that is used to **define** and **configure** Angular **modules**. Its primary purpose is **to provide metadata** that **describes how** (the components, directives, pipes, and services within the module) **are structured, organized, and related to each other**.

3.1 What is the purpose of @NgModule?

First, we need to know what module it is. **A module** is a **cohesive block of code** that **groups** related components, directives, pipes, and services **together**.

**Define a Module**: if you want to create a new Module, needs `@NgModule` syntax.

**Configure Dependency Injection:** Angular's **dependency injection** system relies on modules to organize and provide. using the providers property. (eg: service. By default, services are provided at the root level of the application. If you want to use in this module, you need to use `providers` property of @NgModule. )

( By adding the service to the providers array of a module, you limit its scope to that module and its child modules. Other modules in the application won't have access to this service unless they import the module where the service is provided. )

**Import Other Modules**

**Export :** The @NgModule decorator's **exports** property is used to export these items, making them available for use in other modules.

**Bootstrap the Application**: In Angular applications, there is typically a **root module** (often named AppModule) that serves as the entry point for the application. The @NgModule decorator's **bootstrap** property is used to specify the root component of the application, which Angular will bootstrap when the application starts.

**4. Property Binding**

4.1 What is property binding & explain the difference between interpolation Vs property binding ?

Property binding is **setting the value of an element property using the value of a component property.** ( such as setting the value of an input field, the source of an image, or the disabled state of a button. )

**Syntax**: square brackets ( [] ) in the template.

**Dynamic Updates**: When the value of the component property changes, the corresponding HTML element property is updated automatically. Angular handles this data binding behind the scenes.

**Expression Evaluation**: you can **perform arithmetic operations** or **call functions directly** in the property binding expression.

**Interpolation** is used for **displaying text content** and simple expressions,while **property binding** is used **to bind component properties to HTML element properties for dynamic updates.**

4.2 Property binding with image tag

<img [src]="imageUrl">

<button [disabled]="isDisabled">Click me</button>

**5 Event Binding with click event**

**syntax uses parentheses (())**

**<button (click)="onClick()">Click me</button>**

5.1 Explain the data flow in event binding.

User Interaction --> Event Binding in Template --> Method Invocation --> Data Manipulation --> View Update

----->In the component's template, you use event binding **to listen** for the specific event and **call a method** in the component class **in response to** that event.

<button (click)="onClick()">Click me</button>

----> **invokes** the corresponding method defined in the component class.

----> Inside the method, you can access and manipulate component properties, perform calculations, or interact with external services based on the user action.

----> When a method updates bound component properties, Angular automatically synchronizes the view to reflect the changes, ensuring that real-time UI updates correspond to user activities.

5.2 Difference event binding Vs property binding

**event binding** is specifically designed to handle user interactions in response to those events.

property binding for **setting property values** and event binding for **responding to user events**.

**6 Binding other mouse related events**

mouse-related events such as mouseenter, mouseleave, mousemove, mousedown, mouseup, etc.

<div (mouseenter)="onMouseEnter()" (mouseleave)="onMouseLeave()">Hover over me</div>

**7 Binding keyboard events**

keyboard events such as keydown, keyup, and keypress.

<input type="text" (keydown)="onKeyDown($event)">

**8 $event object**

8.1 Assignment - Checking Shift key status.

<input type="text" **(keydown)="onKeyDown($event)**">

**onKeyDown(event: KeyboardEvent)** {

if (event.**shiftKey**) {

console.log('Shift key is pressed');

} else {

console.log('Shift key is not pressed');

}

}

**9 Two-way Binding (ngModel)**

9.1 What is two-way binding & why do we use it?

It combines property binding and event binding into a single syntax.

Is s commonly used with form elements such as input fields, checkboxes, radio buttons, and selects, where real-time data synchronization between the UI and the component is essential.

9.2 Explain ngModel

ngModel is a directive used for two-way data binding between a form input element (such as <input>, <textarea>, <select>, etc.) and a component property.

<input type="text" [(ngModel)]="name">

first import the FormsModule or ReactiveFormsModule in your Angular module.

9.3 Assignment - (Two-way Binding) Get input field value on button click

<input type="text" **[(ngModel)]**=**"inputValue**">

<button (click)="**getValue()**">Get Value</button>

Component Class (TypeScript):

import { Component } from '@angular/core';

@Component({

selector: 'app-example',

templateUrl: './example.component.html',

styleUrls: ['./example.component.css']

})

export class ExampleComponent {

**inputValue**: string;

**getValue()** {

console.log('Input field value:', this.inputValue);

}

}