**Questions**

1. Chris has created a highlight directive as shown below. What color will be applied to the page if he enters yellow in the input field?\

//highlight.directive.ts

import {Directive, ElementRef,Input} from '@angular/core';

@Directive({

selector:['highlight']

})

export class HighlightDirective{

constructor(el:ElementRef){

    el.nativeElement.style.backgroundColor='blue';

}

}

//app.component.ts

template:`

              <label>Provide color to be applied on the page</label>

              <input type='text' [(ngModel)]='colorName'>

              <div highlight>Hello </div>

1. **color red will be applied to the div element**
2. **color yellow will be applied to the div element**
3. **yellow color will not be applied as there is no binding of model colorName and highlight**

2. What is the correct statement to be placed at line number 3 to disable the button?

Line 1: @Component({

Line 2:             selector: 'my-app',

Line 3:             template: '<button ….>Click Here</button>'

Line 4: })

Line 5: export class AppComponent {

Line 6:          exists: Boolean = true;

Line 7: }

1. **(disabled)="exists"**
2. **disabled = {{ exists }}**
3. **[disabled] = exists**
4. **[disabled]= “exists”**

3. The below code snippet has some lines of code missing.

Line 1: @Component({

Line 2:   template:`

Line 3:                <div>

Line 4:                <h1 [textContent]='"user is" +person.name'> </h1>

Line 5:                <table>

Line 6:                <tr>

Line 7:                <td [attr.colspan]='1+1'>

Line 8:                <button >Female</button>

Line 9:                </td>

Line 10:                       <td [attr.colspan]='1+1'>

Line 11:                       <button>Male</button>

Line 12:                       </td>

Line 13:                       </tr>

Line 14:                       <tr>

Line 15:                       <td>{{person.rating}}</td> //line2

Line 16:                       <td>{{person.address}}</td>

Line 17:                       </tr>

Line 18:                       </table>

Line 19:                       </div>

Line 20:               `

Line 21: })

         export class AppComponent{

             private female={

                              name:'Jenny Wesley' ,

                              gender:'F',

                              rating:4,

                              address:'Wesley states'

                  }

            private male={

                             name:'Ross Green',

                             gender:'M',

                             address:'New York'

            }

            private person=this.female;

        }

As per the requirement mentioned below please complete the lines.

Requirement:

1.set the background color yellow for the div element (line3)

2.hide rating if not present (line15)

1. **line3: [style.backgroundColor]='yellow' line15: <td [hidden]='!person.rating'>**
2. **line3: [style.backgroundColor]=' "yellow" ' line15: <td hidden='!person.rating'>**
3. **line3: [style.background-color]=' "yellow" ' line15: <td [hidden]='!person.rating'>**
4. **line3: [style.background-color]=' "yellow" ' line15: <td hidden='!person.rating'>**

4.In a Library Management application, there is a requirement that all the books should have their price shown in the US-dollar symbol where 'books' is an array containing 'title', 'price', 'dateofpurchase' as attributes. Which of the following is the correct statement to display the price of each book?

1. **<p \*ngFor = "let p of books"> {{ p.price | currency }} </p>**
2. **<p \*ngFor = "let p of books">{{ p.price | currency:"INR" }} </p>**
3. **<p \*ngFor = "let p of books"> {{ p.price | currency:"USD":'code' }} </p>**
4. **<p \*ngFor ="let p of books"> {{p.price | currency:"CAD":'symbol-narrow' }}</p>**

5.What will be the output for template expression with the pipe as given below ?

{{["a", "b", "c", "d", "e", "f"] | slice : 2:-1}}

1. **c,d,e,f**
2. **d,e,f**
3. **c,d,e**
4. **Error: both negative and positive index cant be used together**

6. What will be the output for template expression with the pipe as given below ?

{{'1.25' | percent : '3.3-4'}}

1. **001.250%**
2. **012.500%**
3. **125.000%**
4. **000.125%**

7. Find out the erroneous line in the below code snippet:

Line 1: import { PipeTransform} from '@angular/core';

Line 2: ……

Line 3: export class LengthPipe implements PipeTransform {

Line 4:  length(value: string): string{

Line 5:   return "Length ="+value.length;

Line 6: }

Line 7:}

1. **Line number 1**
2. **Line number 3**
3. **Line number 5**
4. **Line number 4**

8. Sam has created a custom pipe called 'MyPipe' and he wants to use it within a module. In which of the following properties of module metadata, he should declare 'Mypipe'?

1. **Imports**
2. **Providers**
3. **Bootstrap**
4. **Declarations**

9. Find out what is the error  in the below code snippet:

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

……

export class LengthPipe implements PipeTransform {

  length(value: string): string{

  return "Length ="+value.length;

}

}

1. **length method should be an no args method**
2. **LengthPipe class should be using @Component decorator to make it working**
3. **The LengthPipe class has not implemented the transform() method of PipeTransform Interface**
4. **The length method should return an numeric value**

10. In the below code snippet, what is the correct statement to be placed at line number 6 to display the numbers array as a list?

Line 1: import { Component } from '@angular/core';

Line 2: @Component({

Line 3:     selector: 'my-app',

Line 4:    template: `

Line 5:       <ul>

Line 6:        .........

Line 7:       </ul>

Line 8:   `

Line 9:  })

Line 10: export class AppComponent

Line 11: {

Line 12:     numbers: any[] = [1,2,3];

Line 13: }

1. **<li \*ngFor = 'numbers'>{{ numbers }} </li>**
2. **<li \*ngFor = 'num of numbers'>{{ num }} </li>**
3. **<li \*ngFor = '#num of numbers'>{{ num }} </li>**
4. **<li \*ngFor = 'let num of numbers'>{{ num }} </li>**

11. Which of the following are true with respect to directives? (Select two)

1. **A component is also a directive**
2. **There will be no error in the code if you do not write the template property for a directive**
3. **There will be error in the code if you do not write the template property for a directive**
4. **You can leave the template property empty but mentioning template property is must for directives**

12. Which of the following options are correct statements to apply the CSS class 'pink' to the paragraph element? (Select two)

1. **<p [ngClass]='pink'> background is pink</p>**
2. **<p [ngClass]='{pink:true}'> background is pink</p>**
3. **<p [ngClass]='["pink"]'> background is pink</p>**
4. **<p ngClass ={pink:true}>background is pink</p>**

13. In an application, an event when triggered should toggle the element by adding or deleting it from the DOM. When deleting, it should not hide the element in the DOM but has to completely delete it. Which concept in Angular is used to achieve this functionality?

1. **Pipes**
2. **Interpolation**
3. **Structural Directives**
4. **Attribute Directives**

14. Joe has created two components and his requirement is to share data between two components. How he can share data between two components?

1. **He must use @Input and @Output in a component class**
2. **He can directly access one component’s data into another**
3. **He must use id value of each component to share data between them.**
4. **Data sharing between two components is not possible in Angular**

15. What is the correct statement to be placed at line number 5 to load SecondComponent?

**first.component.ts**

Line 1: @Component({

Line 2:              selector:"first-component",

Line 3:              template: `

Line 4:                 <h1> First Component </h1>

Line 5:                   ....

Line 6:              `

Line 7: })

Line 8: export class FirstComponent { }

**second.component.ts**

@Component({

                Selector:"second-component",

                Template: "<h1> Second Component </h1>"

})

export class SecondComponent { }

1. **<SecondComponent></SecondComponent>**
2. **<second-component></second-component>**
3. **SecondComponent.template**
4. **We cannot load SecondComponent**

16. Consider the given code snippets in respective files of ***display***and ***destinations***components.

<!-- display.component.html -->

<div class="col-md-4 offset-4">

    <h2> Best Destinations </h2>

    /\* Line 1 \*/

</div>

/\* destinations.component.ts \*/

destinations = [

    { destinationId: 10, destinationName: "Eiffel Tower", charges: 300 },

    { destinationId: 20, destinationName: "Taj Mahal", charges: 350 },

    { destinationId: 30, destinationName: "Statue of Unity", charges: 250 },

    { destinationId: 40, destinationName: "Great Wall of China", charges: 300 }

]

<!--destinations.component.html-->

<table border="1">

    <thead>

        <tr>

            <th>Destination Name</th>

            <th>Charges</th>

        </tr>

    </thead>

    <tbody>

        <tr \*ngFor="let destination of destinations">

            <td>{{ destination.destinationName }}</td>

            <td>{{ destination.charges }}</td>

        </tr>

  </tbody>

</table>

/\* Line 2 \*/

Which of the following options should be used to render the desired output?

1. **Add <app-display></app-display> in destinations.component.html at Line 2**
2. **Add <app-destinations></app-destinations> in display.component.html at Line 1**

17. Sam wants to create a model driven form for which he should use one of the pre-defined class in the root module. Which class is used for the same?

1. **BrowserModule**
2. **ReactiveFormsModule**
3. **HttpClientModule**
4. **RouterModule**

18. What is the correct statement to bind required and maxlength validators to name field in a reactive form?

1. **name: [' ', Validators.required, Validators.maxlength(10)]**
2. **name: [' ', [Validators.required, Validators.maxlength(10)]]**
3. **name: [Validators.required, Validators.maxlength(10)]**
4. **name: Validators.required, Validators.maxlength(10)**

19. Jack has to create a reactive form in Angular. When end user goes to any input field in the form, he is required to display a popup with some message, irrespective of the fact that user may or may not have typed anything in the field. How can Jack achieve this?

1. **He can use untouched directive.**
2. **He can use touched directive.**
3. **He can use dirty directive.**
4. **He can use pristine directive.**

20.Select the appropriate option to import the EmpService in the module class, so that it would be available to an entire application:

1. **providers: [EmpService]**
2. **imports: [EmpService]**
3. **Inject: [EmpService]**
4. **declarations: [EmpService]**

21. Find the correct statement to be placed at Line number 1 to create a custom service?

Line 1:     .....

Line 2:  export class MyService {

Line 3:      books: any[] = [{id:1,name:'TypeScript'},{id:2, name:'Angular'}];

Line 4:      getData(){

Line 5:         return books;

Line 6:      }

Line 7:  }

1. **@Service({name:'myService'})**
2. **@Injectable()**
3. **@inject()**
4. **@Injectable({name:'myService'})**

22. Pick the correct statements regarding services in Angular. (Select 3 options)

1. **Services can be used for network connection tasks.**
2. **Services can be used for writing reusuable templates for components.**
3. **Services can be used for writing reusable operational tasks for components.**
4. **Angular will take care of injecting the mentioned service dependencies.**

23. Consider the following code snippets created for an Angular application.

**Employee.ts:**

export class Employee {

  constructor() {  }

}

Assume you have defined two variables of class Employee type in our component as given below.

emp1 = new Employee();

emp2 : Employee;

What will happen when the below code is executed?

<div \*ngIf="emp1">

  Inside first div element

</div>

<div \*ngIf="emp2">

  Inside second div element

</div>

Assumption(s): All the components are registered in the app component of the application.

1. **Only <strong>Inside first div element</strong> is displayed**
2. **Only <strong>Inside second div element</strong> is displayed**
3. **Both <strong>Inside first div element<br />Inside second div element</strong> are displayed**
4. **The code results in an error because \*ngIf directive can be used only once in an Angular 6 component.**

24. Consider the following code snippet.

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

@Component({

    selector: 'my-app',

    templateUrl: 'compute.component.html'

})

export class ComputeComponent {

    items = [5,10,15,20];

}

Identify the appropriate code snippets that should be used in 'compute.component.html' so that the output will be as follows:

Odd Number

Even Number

Odd Number

Even Number

1. **<div \*ngFor="let item of items"> <div \*ngIf="item % 2 == 0"> <div>Even Number</div> <div>Odd Number</div> </div></div>**
2. **<div \*ngFor="let item of items"> <div \*ngIf="item % 2 == 0"> <div>Even Number</div> </div> <div \*ngIfElse="item % 2 == 1"> <div>Odd Number</div> </div> </div>**
3. **<div \*ngFor="let item of items"> <div \*ngIf="item % 2 == 0"> <div>Even Number</div> </div> <div>Odd Number</div> </div>**
4. **<div \*ngFor="let item of items"> <div \*ngIf="item % 2 == 0"> <div>Even Number</div> </div> <div \*ngIf="item % 2 == 1"> <div>Odd Number</div> </div> </div>**

25. The following error occurred while executing the following code snippet. Identify the appropriate options to fix the errors

Error: **Can't bind to 'ngFor' since it isn't a known property of 'tr'**

<tr \*ngFor="city in cities">

    <td>city</td>

</tr>

**Note:** You are expected to display the values instead of index.

1. **Replace the expression as follows: <tr \*ngFor="city of cities"> <td>city</td></tr>**
2. **Replace the expression as follows: <tr \*ngFor="let city in cities"> <td>city</td></tr>**
3. **Replace the expression as follows: <tr \*ngFor="var city in cities"> <td>city</td></tr>**
4. **Replace the expression as follows: <tr \*ngFor="let city of cities"> <td>city</td></tr>**

26. Which among the following is/are a valid syntax for ngif expression?

 1. <div \*ng-If="your-condition">...</div>

 2. <div \*ngIf="your-condition">...</div>

 3. <div template="ngIf your-condition">...</div>

 4. <ng-template [ngIf]="your-condition">  
    <div>...</div>  
    </ng-template>

1. **Only 2 and 3 are valid**
2. **Only 2,3 and 4 are valid**
3. **Only 2 is valid**
4. **Only 1 is valid**

27. Potter has written the below code where status has value as false. He and his friend are arguing on the output. Consider the below code and decide who is write among them.

<div [ngIf]="status">

  <h2>Hello John </h2>

</div>

<div [hidden] = "!status">

<div>

  <h2> Hello Alice</h2>

</div>

</div>

A. John Says output could be “Error [ngIf] should be \*ngIf”

B. Alice says output could be “Hello Alice”

1. **John is Correct**
2. **Alice is Correct**
3. **John and Alice both are Correct**
4. **Both are wrong**