**Lesson29 Singleton by Applying Dependency Injection**

**Notes:-**

**1-we will make the following example about applying the standard way without dependency injection and another example with applying dependency injection**

**A-Without Applying Dependency Injection**

**1-in HomeComponent.html we will write the following code**

<p>

home works!

</p>

<input type='text' [(ngModel)]='colour' [style.background]="colour"/>

**2-in HomeComponent.ts we will write the following code**

//we see that in this way every component has it’s local instance and the instance in each load it’s reset because it’s create new instance

private \_testServie:TestServiceService;

constructor(){

this.\_testServie = new TestServiceService();}

ngOnInit() {}

get colour(): string {

return this.\_testServie.SelectedColor;}

// Implement a setter to change the colourPreference value

// of the service

set colour(value: string) {

this.\_testServie.SelectedColor = value;}

<p>

home works!

</p>

<input type='text' [(ngModel)]='colour' [style.background]="colour"/>

**2-In Home2Component.ts we will write the following code**

<p>

home2 works!

</p>

<input type='text' [(ngModel)]='colour' [style.background]="colour"/>

**And in Code Behind we will type**

private \_testServie:TestServiceService;

constructor(){

this.\_testServie = new TestServiceService();}

ngOnInit() {}

get colour(): string {

return this.\_testServie.SelectedColor;

}

// Implement a setter to change the colourPreference value

// of the service

set colour(value: string) {

this.\_testServie.SelectedColor = value;}

**B-Applying Dependency Injection**

**1-In HomeComponent.html we will type the following code**

<p>

home2 works!

</p>

<input type='text' [(ngModel)]='colour' [style.background]="colour"/>

**And In Code Behind**

constructor(private \_testServie:TestServiceService){}

ngOnInit() {}

get colour(): string {

return this.\_testServie.SelectedColor;

}

set colour(value: string) {

this.\_testServie.SelectedColor = value;}

**2-In Home2Component.html we will type the following code**

<p>

home works!

</p>

<input type='text' [(ngModel)]='colour' [style.background]="colour"/>

private \_testServie:TestServiceService;

constructor(private \_testServie:TestServiceService){

this.\_testServie = new TestServiceService();}

ngOnInit() {}

get colour(): string {

return this.\_testServie.SelectedColor;}

set colour(value: string) {

this.\_testServie.SelectedColor = value;}