Angular5 Interview Questions

Virendra Pandey

M.Sc. in Information Technology

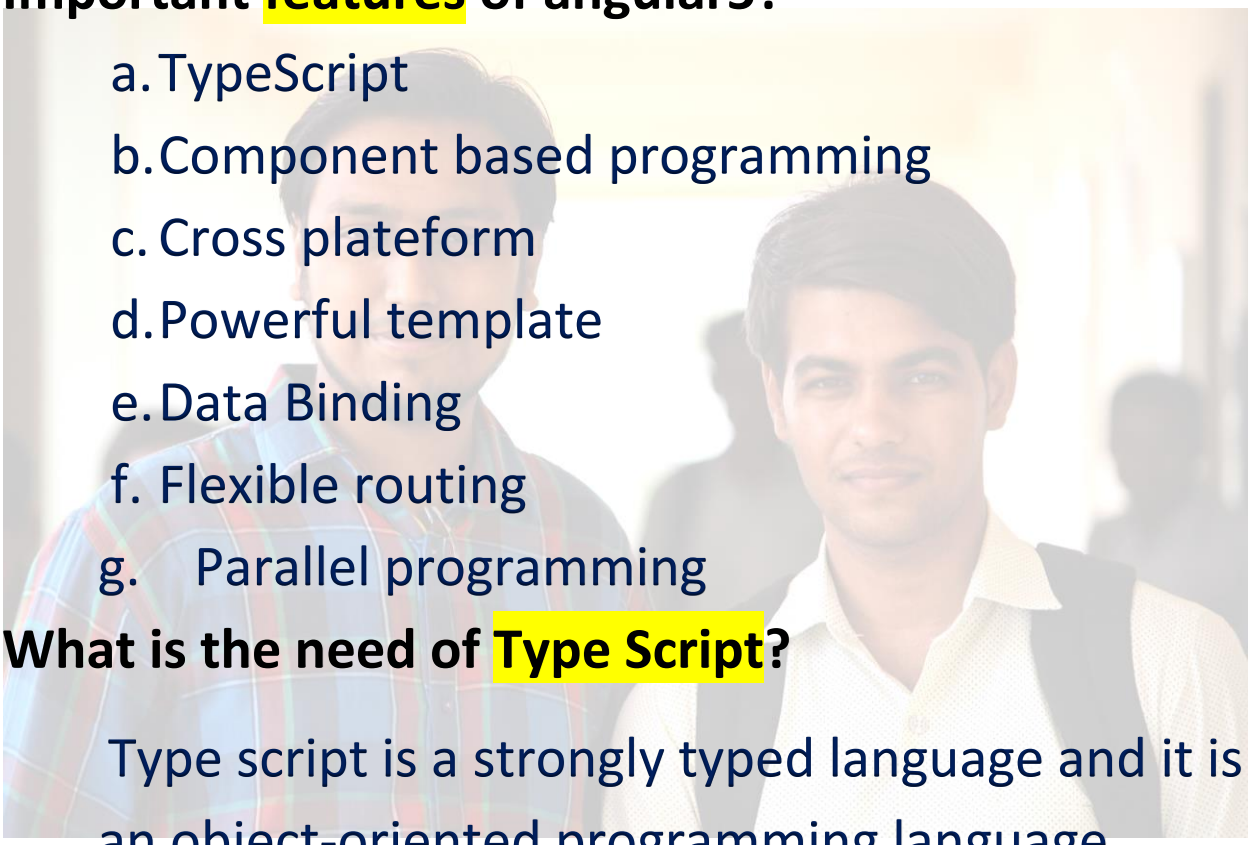
Angualr5 Interview Question

1. **What are the key components of Angular5?**

   Key component means primary programming items are Module, Component, Services and Pipes.

2. **Important features of angular5?**

   a. TypeScript
   b. Component based programming
   c. Cross plateform
   d. Powerful template
   e. Data Binding
   f. Flexible routing
   g. Parallel programming



3. **What is the need of Type Script?**

   Type script is a strongly typed language and it is an object-oriented programming language. Angular uses lots of type script features for lots of its core functionalities.

4. Deference between Type Script and java Script?

S.N.	Type Script	Java Script
1.	Type script is a strongly typed language.	Java Script is not a typed language.
2.	Type script is an object-oriented programming language. And support all oops concepts.	Java Script a Procedure-Oriented programming Language and also it is not support all oops concepts.
3.	It has fully compilation than execution.	It has only execution.
4.	Programming errors can be find out at compile time	Programming errors cannot be identified.
5.	Type Script doesn't run in browser.	JavaScript run in browser.
6.	Type script is superset of java script.	Java script is subset of ECMA script 6.
7.	It supports Interfaces , generics, Strongly typing etc…	Java script doesn't support.

5. What is angular CLI?

Angular CLI stands for "Command Line Interface". Angular CLI helps to create new projects, components, pipes, services, modules and build a

project easily. It saves lots of efforts to generate new items of Angular.

6. How to create new project?

Angular5 uses a CLI command to create a new project:

```
D:\>Angular5>ng new project_name
```

7. How to add Feature Module?

Module using CLI:

```
D:\>Angular5>ng g module module_name
```

8. How to add new Component?

Component using CLI:

```
D:\>Angular5>ng g component component_name
```

9. How to build new project?

Build a new project using CLI:

```
D:\>Angular5>ng g build -prod
```

10. Explain Modules in Angular5?

Angular module is a logical group of required programming items that are need to execute application.

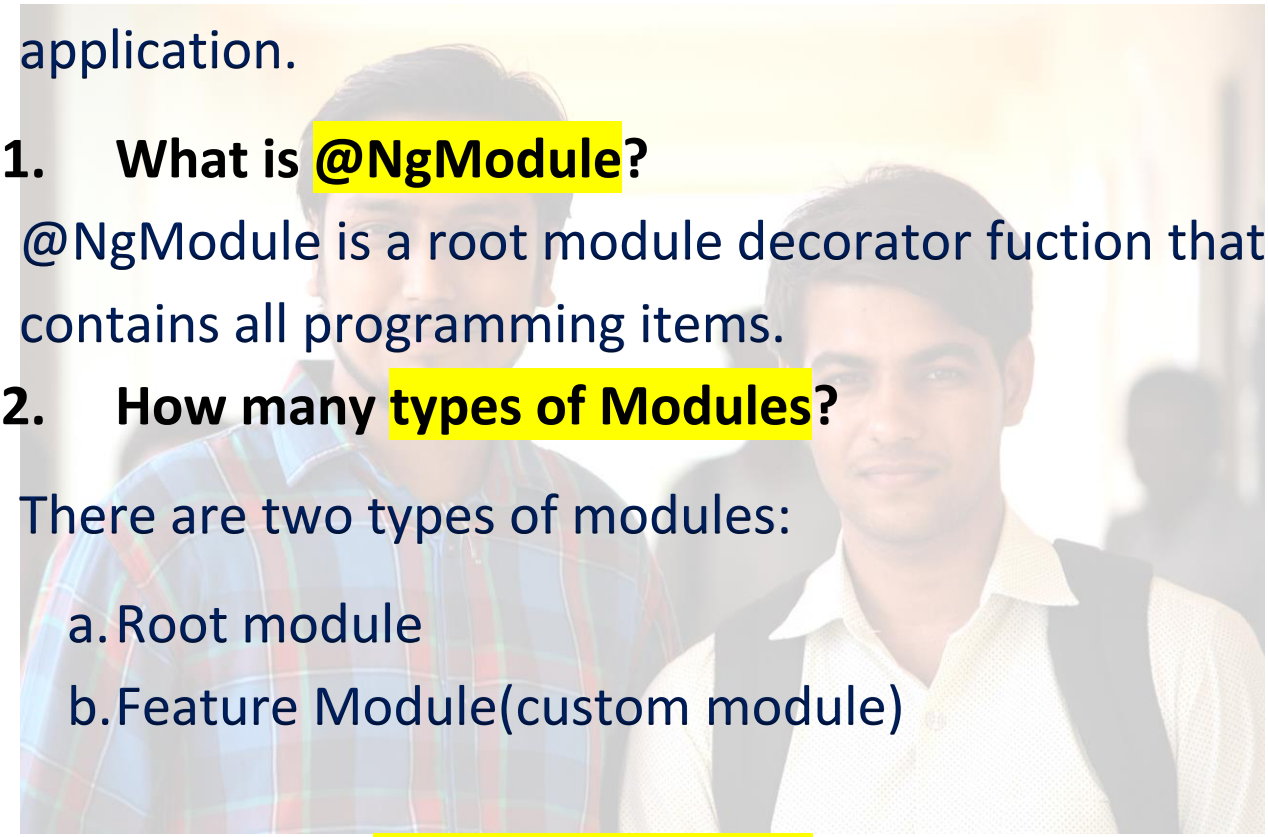
11. What is @NgModule?

@NgModule is a root module decorator fuction that contains all programming items.

12. How many types of Modules?

There are two types of modules:

a. Root module
b. Feature Module(custom module)



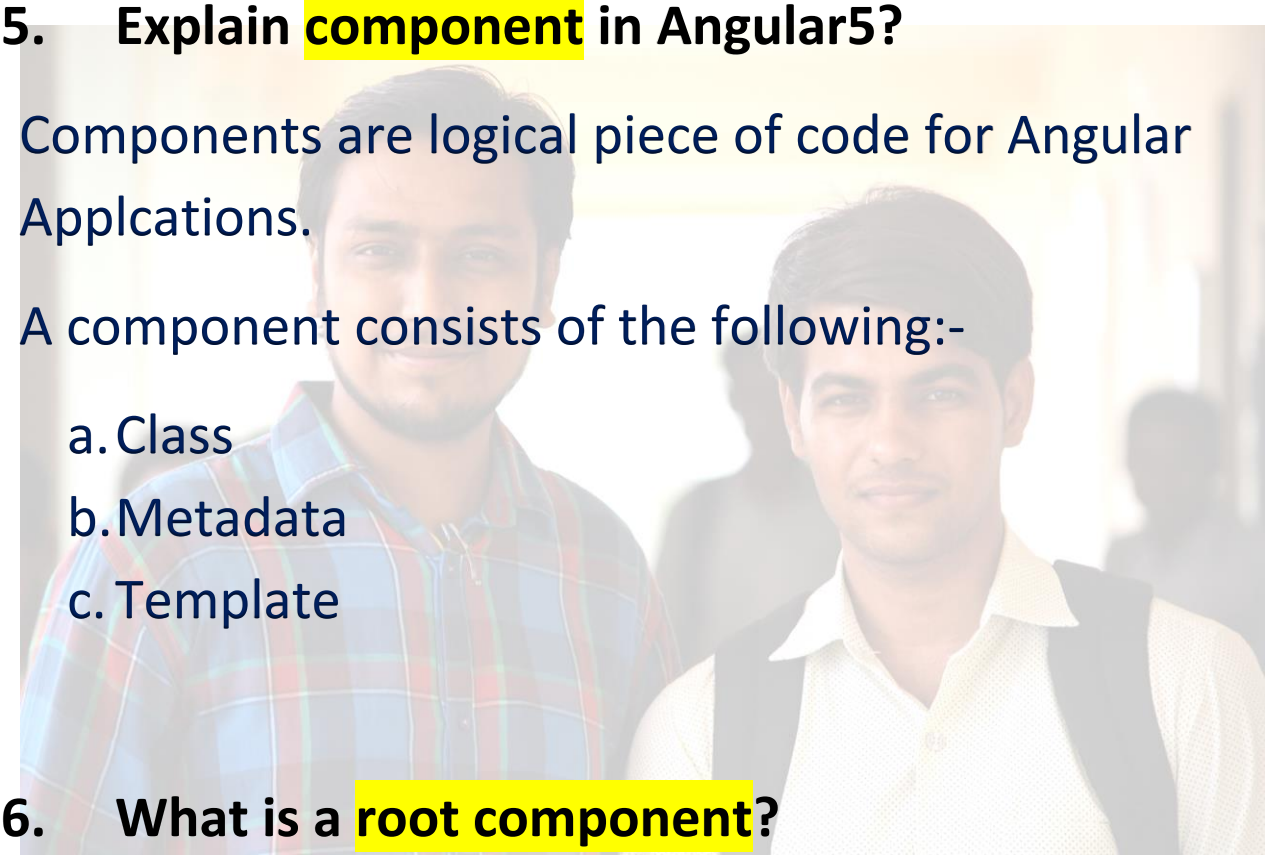
13. Which file bootstrap module?

main.ts file bootstrap module.

14. What is feature module?

When our root module start growing, it starts to be evident that some elements (components, directives, etc.) are related in a way that almost feel like they belong to a library that can be "plugged in".

15. Explain component in Angular5?



Components are logical piece of code for Angular Applcations.

A component consists of the following:-

a. Class
b. Metadata
c. Template

16. What is a root component?

Root component is app.component.ts file which bootstrapped in app.module.ts file.

17. Why do we use decorator?

Component decorator allows you to mark a class as an Angular component and provide additional metadata that determines how the component should be processed, instantiated and used at runtime.

18. Explain metadata of component?

When you configure a component for example, you're providing metadata for that class that tells Angular that we have a component, and that component has a specific configuration. Like:

Selector, providers, styles, styleUrls, template, templateUrls etc…

19. What is @Input and @Output?

@Input decorator allow you to pass the data from parent component to a child component

@Output decorator allow you to send data from child component to a parent component.

20. What is template? Why do we use?

Templates are the html files that is used to preparing presentation logic in order to present component data to end-user.

21. What is template variable?

Template variables are the special variables which are refers DOM object. It is declared using special symbol hash(#) like: <input type="text" #t>

22. When do we use { }, [ ], ( ), [()]?

{ }, [ ], ( ), [()] are used data binding concept.

{}:- it is used for one-way data binding {{uname}}

[]:- it is used for property binding {{hidden}}

():- it is used for event binding (click)

[()]:- it is used for two-way data binding [(ngModel)]

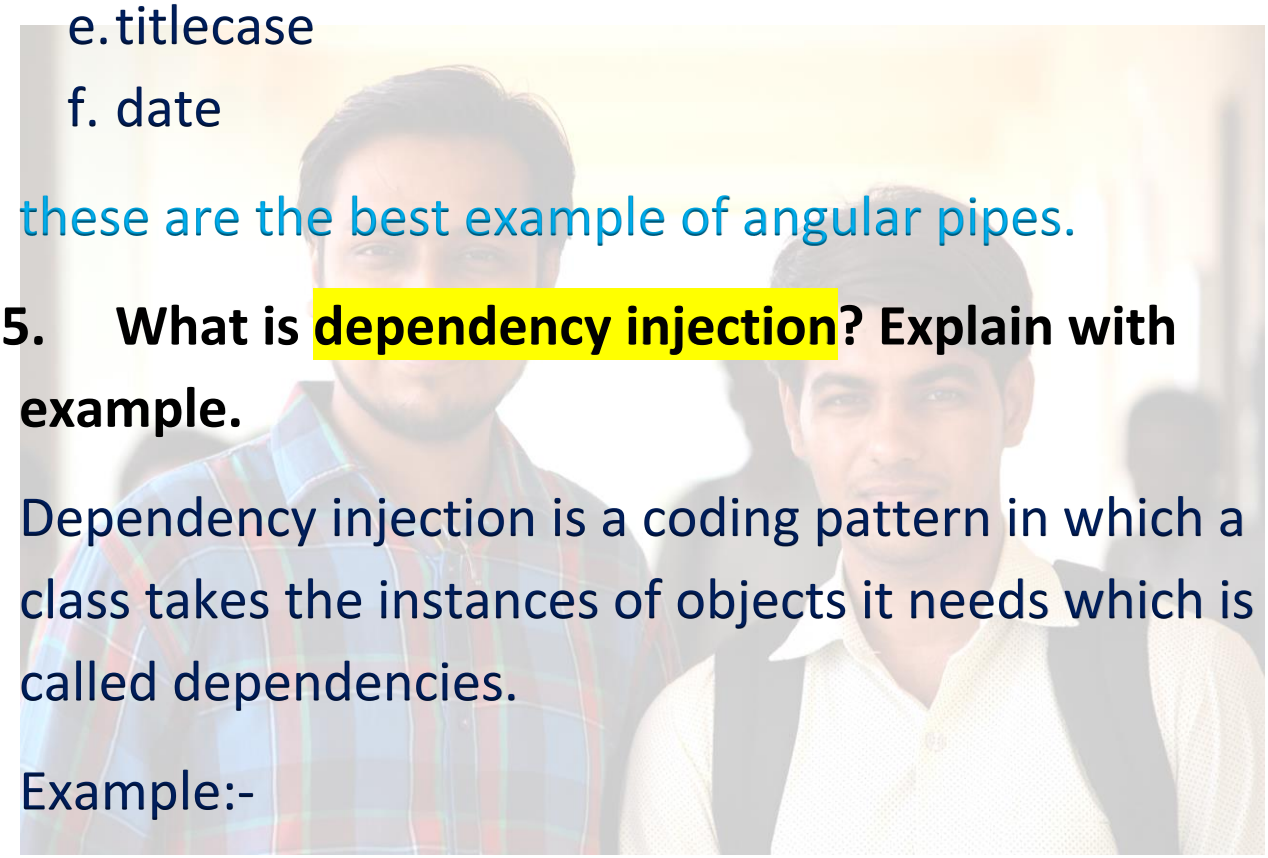
23. Explain the concept of pipes?

Pipes are used to transforming the data in the template itself. We can convert component data in customized format like uppercase, lowercase…

24. What are the examples of pipe?

a. uppercase
b. lowercase
c. currency
d. slice
e. titlecase
f. date

these are the best example of angular pipes.



25. What is dependency injection? Explain with example.

Dependency injection is a coding pattern in which a class takes the instances of objects it needs which is called dependencies.

Example:-

1. demo.component.ts

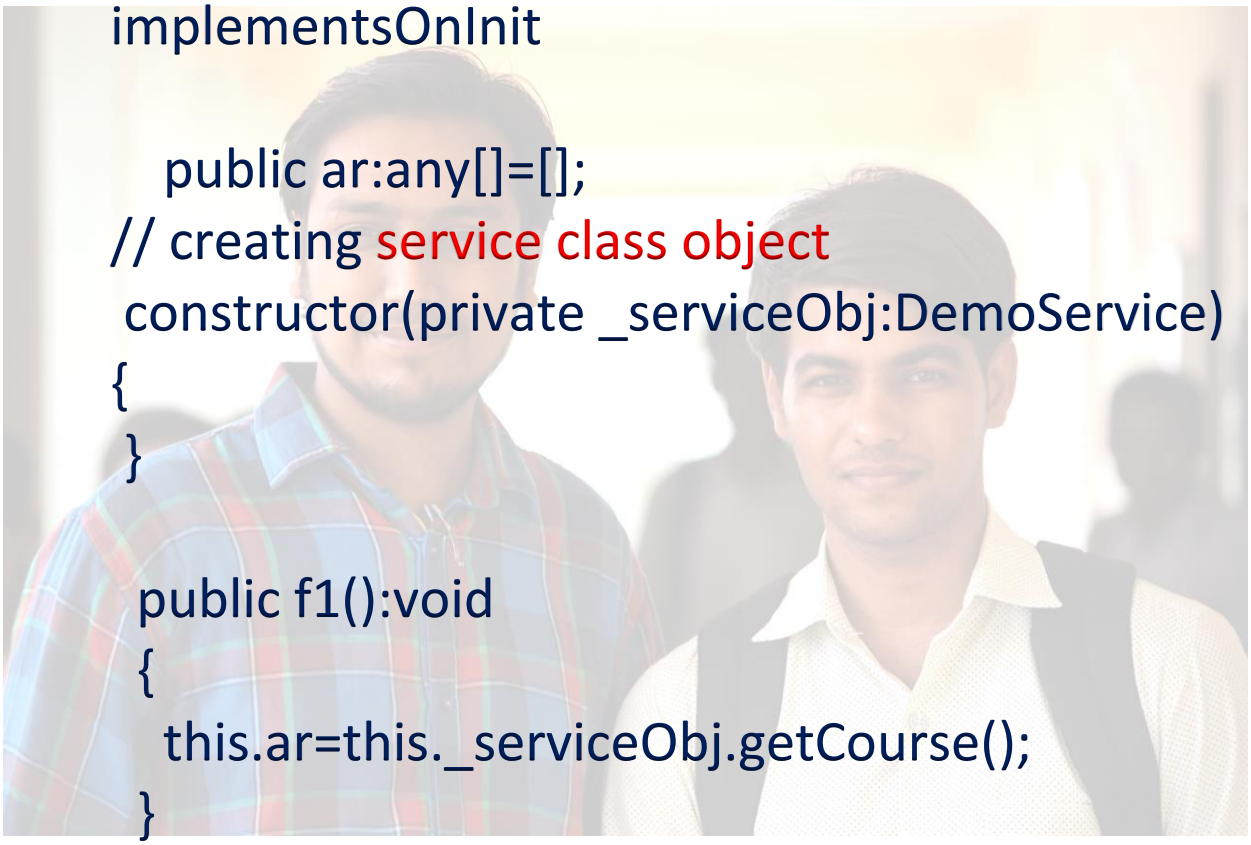
```
import { Component, OnInit } from
'@angular/core';
import { DemoService } from '../demo.service';
```

```
@Component({
  selector: 'app-course',
  templateUrl: './course.component.html',
  styleUrls: ['./course.component.css']
})
export class CourseComponent
implementsOnInit

    public ar:any[]=[];
// creating service class object
 constructor(private _serviceObj:DemoService)
{
 }

  public f1():void
  {
    this.ar=this._serviceObj.getCourse();
  }
  ngOnInit() {
  }

}
```



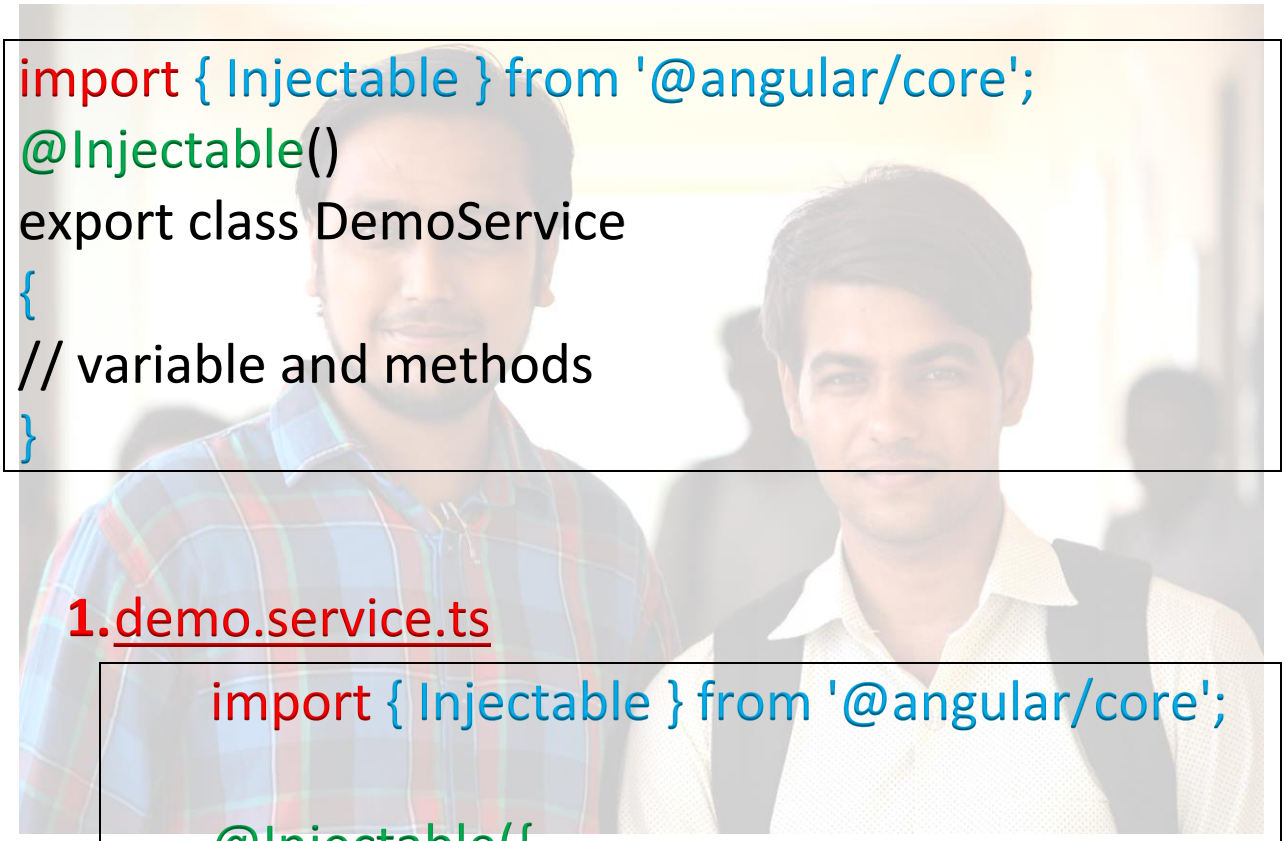
26. How to create and inject services in component?

We can create a service using CLI command

```
ng g service service_name
```

Syntax:

```
import { Injectable } from '@angular/core';
@Injectable()
export class DemoService
{
// variable and methods
}
```

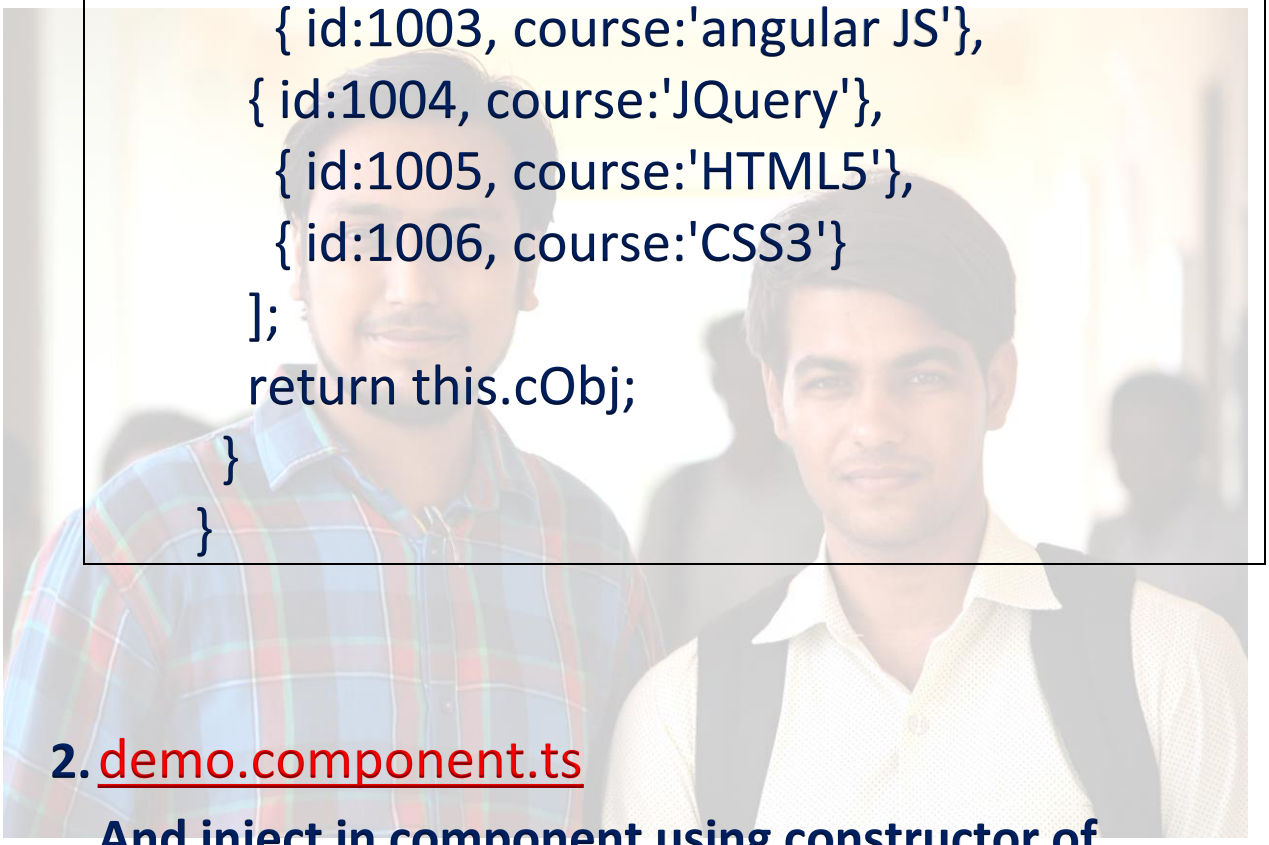


1. demo.service.ts

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class DemoService
{
   public cObj : any[] = [];
```

```
constructor() { }
        public getCourse():any[]
 {
   this.cObj = [
     { id:1002, course:'angular5'},
     { id:1003, course:'angular JS'},
    { id:1004, course:'JQuery'},
     { id:1005, course:'HTML5'},
     { id:1006, course:'CSS3'}
    ];
    return this.cObj;
  }
}
```



2. demo.component.ts

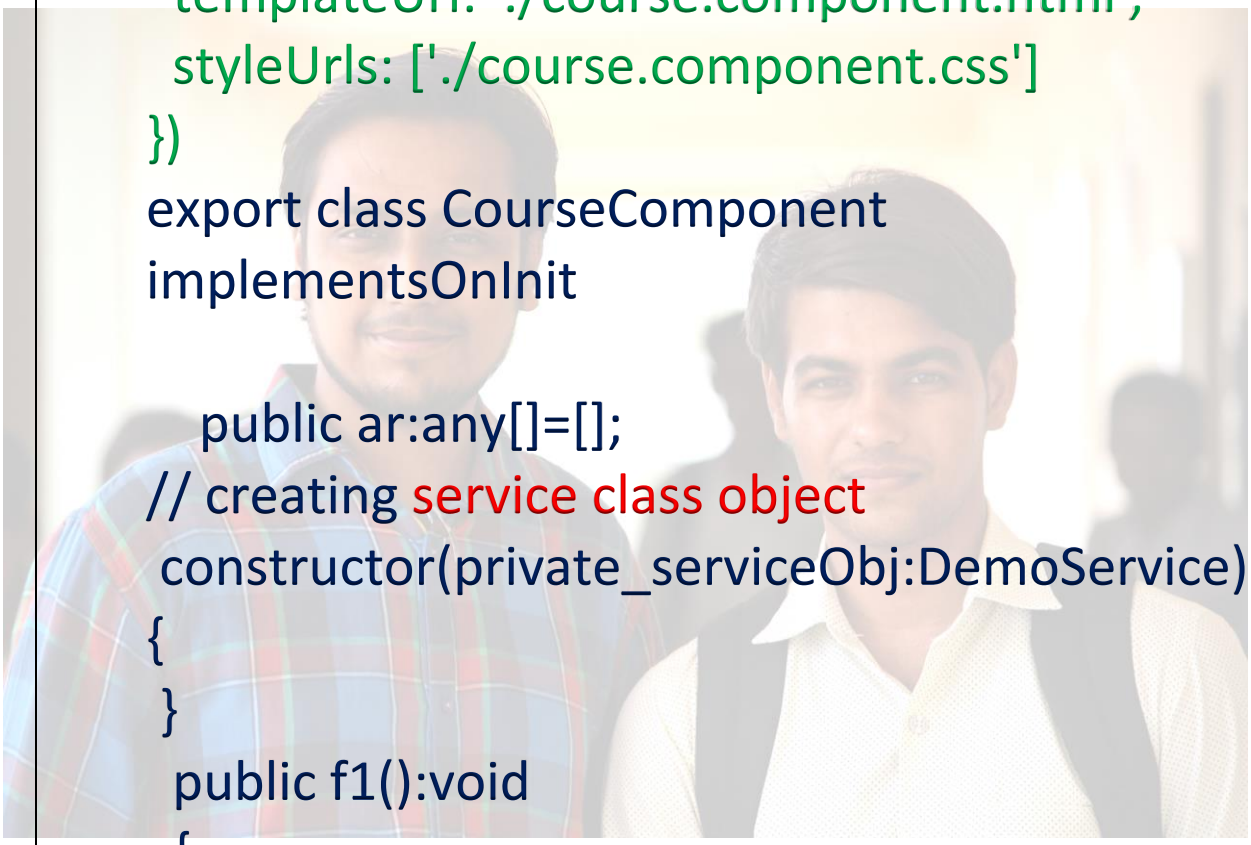
And inject in component using constructor of component class:

```
Constructor(private _httpClient : HttpClient)
 {


   }
```

```
import { Component,OnInit} from '@angular/core';
import { DemoService } from'../demo.service';
@Component({
      selector: 'app-course',
      templateUrl: './course.component.html',
      styleUrls: ['./course.component.css']
    })
    export class CourseComponent
    implementsOnInit

      public ar:any[]=[];
    // creating service class object
     constructor(private_serviceObj:DemoService)
    {
     }
      public f1():void
      {
        this.ar=this._serviceObj.getCourse();
      }
      ngOnInit() {
      }
```



```
}
```

27. Which service is used to communicate with server?

HttpClient service is used to communicate with server. It belongs to @angular/common/http library.

28. What is ngOnInit?

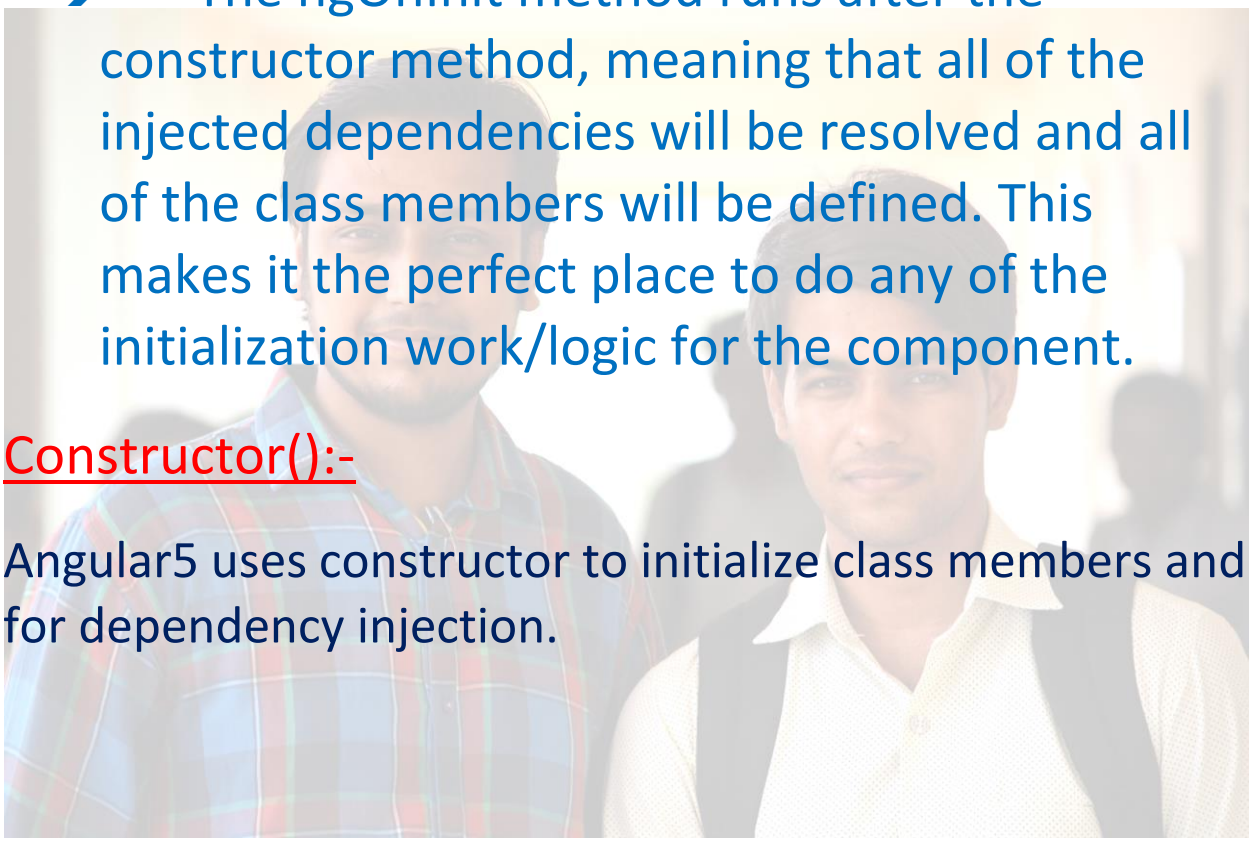
➔ Initialize the directive/component after Angular first displays the data-bound properties and sets the directive/component's input properties.

➔ The ngOnInit method runs after the constructor method, meaning that all of the injected dependencies will be resolved and all of the class members will be defined. This makes it the perfect place to do any of the initialization work/logic for the component.

29. What is the difference between ngOnInit and constructor

ngOnInit():-

- ➔ Initialize the directive/component after Angular first displays the data-bound properties and sets the directive/component's input properties.
- ➔ The ngOnInit method runs after the constructor method, meaning that all of the injected dependencies will be resolved and all of the class members will be defined. This makes it the perfect place to do any of the initialization work/logic for the component.



Constructor():-

Angular5 uses constructor to initialize class members and for dependency injection.