**Note:-Use Code shown in class for Understanding.**

**What is Dependency Injection?**

**Dependency Injection** (DI) is a software design pattern that deals with how components get hold of their **dependencies**. The **AngularJS** injector subsystem is in charge of creating components, resolving their **dependencies**, and providing them to other components as requested.

Dependency injection has two principles

1. Single Responsibity--only 1 service at a time.

2. Dependency Inversion--hardcoded values are not passed.

Dependency Injection is a software design in which components are given their dependencies instead of hard coding them within the component. It relieves a component from locating the dependency and makes dependencies configurable. It also helps in making components reusable, maintainable and testable.

**What is need of Dependency?**

Dependency Injection is Software Design Pattern.

2.Components are given there dependencies and no values

3. Make Dependency Configurable

4. Makes component Reusable, Maintainable and Testable

**DI in angular JS**

1. Angular JS has built in Dependency Mechanism.

2. Allows to divide application into smaller Modules

3. Smaller modules can be injected into each other

4. Makes the modulatization easier

**How to Inject Dependency Into Applicatio**

AngularJS provides a supreme Dependency Injection mechanism. It provides following core components which can be injected into each other as dependencies.

* Value
* Factory
* Service
* Provider
* Constant
* Value

Value is a simple JavaScript object, which is required to pass values to the controller during config phase (config phase is when AngularJS bootstraps itself).

## Factory

Factory is a function which is used to return value. It creates a value on demand whenever a service or a controller requires it. It generally uses a factory function to calculate and return the value.

## Service

Service is a singleton JavaScript object containing a set of functions to perform certain tasks. Service is defined using service() function and it is then injected into the controllers.

Constant

Constants are used to pass values at the config phase considering the fact that value cannot be used during the config phase.

mainApp.constant("configParam", "constant value");

## Provider

Provider is used by AngularJS internally to create services, factory, etc. during the config phase. The following script can be used to create MathService that we created earlier. Provider is a special factory method with get() method which is used to return the value/service/factory.

//define a module

var mainApp = angular.module("mainApp", []);

...

//create a service using provider which defines a method square to return square of a number.

mainApp.config(function($provide) {

$provide.provider('MathService', function() {

this.$get = function() {

var factory = {};

factory.multiply = function(a, b) {

return a \* b;

}

return factory;

};

});

});