→ Inversion of control ⇒ IoC Container

→ Dependency Injection

→ TicTacToe

→ Parkinglot]

↓

models

repo

service ←

controller

create
a parkinglot

↓

Dio → controller → service → repo

↓
injecting
the service obj
in constructor

injecting
the repo. obj
in constructor

Parkinglot ⇒ developer is responsible for managing dependencies.

⇒ IoC ⇒ The control over dependency injections [create & inject] is taken away from "main functional code" and is done by the framework / language that is being used.

⇒ <u>IoC Container</u> ⇒ part of Spring that actually takes care of dependency injection in Spring.

⇒ main class/ starting class is responsible for creating all the required objects, but it is inverted in Spring,

⇒ IoC Container is responsible for managing bean lifecycle :-

↓

BEAN is objects required for running the app^n

⇒ <u>Bean lifecycle</u>

i) Instantiation ⇒ very first step is object creation

ii) Populate the properties ⇒

⇒ injecting dependent objects ⇒

- Constructor Injection
- Setter Injection
- Field/ Attribute Injection

iii) Set Bean names

iv) Beanfactory ⇒ map the bean names & objects.

v) Pre-Initialization → Initialization → Post. Initialization

vi) Bean is ready

vii) Bean destruction.

Pre-Initialization → Injecting the objects in required attribute field

↓

Initialization → getting the ready to use Object

↓

post Initialization ⇒ take care of post initialization steps / tasks.

→ everyday ⇒ tasks to complete

⇒ best performer & worst performer

⇒ 1:1s [60% attendance & psp]

→ 16/10 starts. 12/10 off

Short break @ <u>8:25AM</u>