

Class → 11

Agenda →

- 1. IOC [Inversion of Control]
- 2. DI [Dependency Injection]
- 3. Beans
- 4. Annotations
- 5. MVC (Model View Controller)

Scenario →

1. You want to commute to your ofc.

(a) Drive via your car

(b) Someone else drop you to ofc everyday

Object Creation

① You can create a JAVA obj yourself
(new)

② Spring will do that job for you

We want Spring to handle the obj lifecycle
(Initialisation, deinitialisation etc)

① IOC →

C1
C2
C3
C4
C5

Bean → ~~class~~
→ Object
object which spring will create

@ component

(IOC) → phenomenon by which Spring

creates an obj for us instead of we manually creating that obj

IOC containers / Spring containers / Spring context

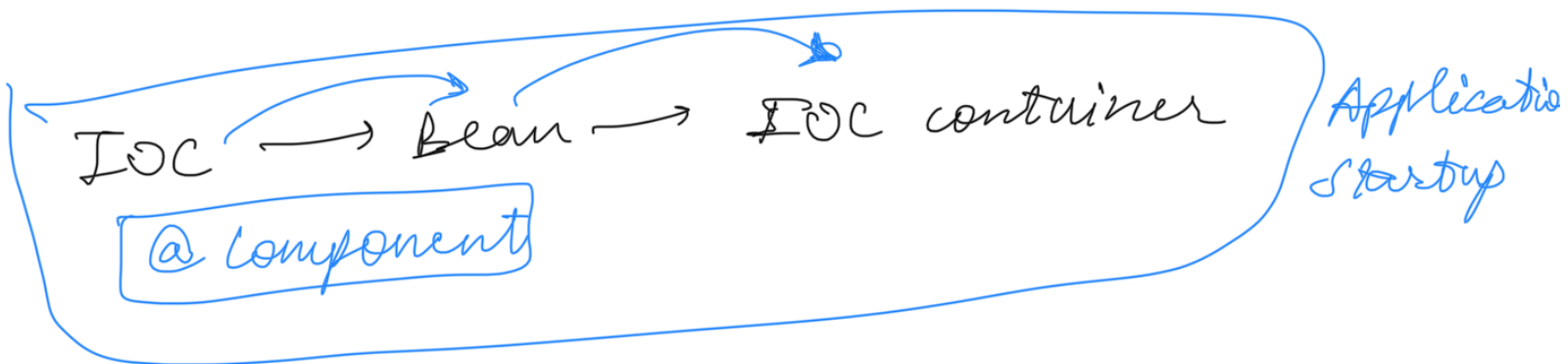
Memory

Application context

Beans which are created by spring will be stored here

Synchronised
Critical section
}

~~HashMap~~ → X



100 classes

P1
→ C1-C50
P2
→ C51-C100

10-15 → @component
compiled

MVC

Model View Controller

Design pattern

Java Framework

→ Dispatcher Servlet to accept requests

Map
Right resources

(MVC)

Controller, Model & View

