

# SPRING CORE:

- **SpringBoot Intro**

1. @SpringBootApplication = @ComponentScan + @EnableAutoConfiguration
2. SpringBoot uses ApplicationContext by default and the run method(SpringBoot.run) returns object of type ApplicationContext.
3. There are 52 beans behind the scene to support this autoconfiguration by the springboot.

- **Spring Bean Lifecycle:**

```
5 public class Greeting
7 {
3°   static
3   {
3       System.out.println("Static Block");
1   }
2
3°   {
4       System.out.println("Non static Java init Block");
5   }
5
7°   public Greeting()
3   {
9       System.out.println("Greeting bean created --> Constructor");
2   }
```

- We have one static block, non-static (or java initialization) block and greeting method.



:: Spring Boot :: (v3.3.5)

```
2024-11-05T11:00:19.351+05:30 INFO 23904 --- [SpringBootApplication] [
2024-11-05T11:00:19.352+05:30 INFO 23904 --- [SpringBootApplication] [
```

Static Block

Non static Java init Block

Greeting bean created --> Constructor

```
2024-11-05T11:00:20.325+05:30 INFO 23904 --- [SpringBootApplication] [
```

- Static → Non Static → Constructor.

Spring Bean Lifecycle | Spring Core Part 7 | Spring Boot Tutorial | Spring Framework

```
24 @PostConstruct
25 public void init()
26 {
27     System.out.println("Bean init method");
28 }
29
30 public String generateWish()
31 {
32     return "Good Night";
33 }
34
35 @PreDestroy
36 public void destroy()
37 {
38     System.out.println("Bean destroyed");
39 }
40 }
```

Console: <terminated> SpringBootApplication [Java Application] C:\Program Files\Java\jdk-17\bin\java.exe (05-Nov-2024, 11:01:51 am)

- The init method has @PostConstruct due to which it will be called right after constructor calling.
- The destroy method has @PreDestroy due to which it will be called after whole execution of program (at the end).