

## Java Bootcamp – Day 19 - Exercise (Spring Container)

**Submitted by: Nawaf AlMutairi**

**Date: 13/11/2024**

### Contents

1. Question 1
2. Question 2
3. Question 3
4. Question 4
5. Question 5

### Question 1:

**Output: (Only one case)**

hey from message1

**Explanation:**

We only have 1 method 'getMessage1()' and it should be working well, because we used @Bean annotation, and it prints the message then returns "1".

### Question 2:

**Output: (Only one case)**

hey from message1

hey from message2

**Explanation:**

Message1 method printed before Message2, because @Qualifier annotation has been used, so Message2 depends on Message1's return value, and it can't be executed before Message1.

### Question 3:

**Output: (Three Cases)**

**Case 1:**

hey from message1

hey from message3

hey from message2

**Case 2:**

hey from message3

hey from message2

hey from message1

**Case 3:**

hey from message3

hey from message1

hey from message2

**Explanation:**

Message2 can NOT be executed before Message3 because of `@Qualifire` annotation, however Message1 does not depend on any method, so it can be executed anytime.

**Question 4:****Output (Six Cases)****Case 1:**

hey from message1

hey from Main controller

hey from message3

hey from message2

**Case 2:**

hey from message 3

hey from message2

hey from message1

hey from Main controller

**Case 3:**

hey from message1

hey from message3

hey from message2

hey from Main controller

**Case 4:**

hey from message1

hey from message3

hey from Main controller

hey from message2

**Case 5:**

hey from message3

hey from message1

hey from message2

hey from Main controller

**Case 6:**

hey from message3

hey from message1

hey from Main controller

hey from message2

**Explanation:**

Because of the unexpected context of Spring, we cannot expect exactly what the output is, but as we can see, Message2 can NOT be executed before Message3, and MainController can NOT be executed before Message1, and that is because of @Qualifier annotation, Message2 depends on return value of Message3, and MainController depends on return value of Message1. Other than that, Springboot behavior decides the order of methods.

**Question 5:**

**Output: (Only one case)**

hey from message3

hey from message2

hey from Main controller

hey from message1

**Explanation:**

There is only 1 case, and the reason is that Message1 depends on MainController, and MainController depends on Message2, and Message2 depends on Message3. And that is because of @Qualifier annotation. Message1 receives MainController object, but the constructor of MainController receives a String from Message2, and Message2 receives a String from message3.