**Answer Q1:**

**The output of: hey from massage 1**

**This phrase" hey from massage 1 " will be printed because it's a single method and it has been called.**

**Answer Q2:**

**The output of:**

**hey from massage 1**

**hey from massage 2**

**Since the method getMessage2() depends on the result of getMessage1() using @Qualifier("1"), it executes after getMessage1(), leading to printing "hey from massage 2".**

**Answer Q3:**

**hey from massage 1**

**hey from massage 3**

**hey from massage 2**

**The reason for this is that the method getMessage2() depends on the result of the method getMessage3(), which is defined using @Qualifier("3"). When specifying this, Spring searches for the bean defined with @Qualifier("3"), finds it, and runs it to use its value in getMessage2(), leading to printing "hey from massage 3" before "hey from massage 2".**

**Answer Q4:**

**hey from massage 1**

**hey from Main controller**

**hey from massage 3**

**hey from massage 2**

**The reason is that the method getMessage 1() is annotated with @Qualifier("1") and is printed first. Then, the MainControllar component is initialized and printed. After that, the method message3() is executed and printed. Finally, the method getMessage 2() is executed and printed.**

**Answer Q5:**

**hey from massage 3**

**hey from massage 2**

**hey from Main controller**

**hey from massage 1**

**The reason for this output is the order in which the methods are defined and executed in the code.**

**First getMassge3() method is defined and executed, printing "hey from massage 3". Then, getMassge2() method is executed, printing "hey from massage 2". After that, the MainControllar component is initialized using the return value from getMassge2(), printing "hey from Main controller". Finally,**

**getMassge1() methods is excuted, printing " hey from massage 1".**