Question1:

-Possible output1 :

```
hey from message1
```

Explenation:

There are no dependencies and only one method in this class.

Question2:

-Possible output1 :

```
hey from message1
hey from message2
```

Explenation:

Since there is @Qualifier annotation ("1") before the "getMessage1" method, and this qualifier exists before the "getMessage2" parameter, the "getMessage2" method depends on the "getMessage1" method and cannot run before it.

Question3:

-Possible output1 :

```
hey from message1
hey from message3
hey from message2
```

Explenation:

Since "getMessage1" method doesn’t have any parameter/ doesn’t depend on other method, its print statement will print first; after that, the same approach for "getMessage3" method because it doesn’t have any parameter/ doesn’t depend on other method, its print statement will print the second one; the last one is "getMessage2" method because it’s depends on "getMessage3" method and can’t run before it.

****

-Possible output2 :

```
hey from message3
hey from message2
hey from message1
```

Explenation:

Since the "getMessage3" method is the same as the "getMessage1" method, which doesn’t have any parameters and doesn’t depend on the other method, there is also another possibility to print its print statement first; after that, the "getMessage2" method will run because this method depends on the "getMessage3" method due to the presence of a qualifier ("3"), and lastly, the "getMessage1" method’s will print its print statement.

****

-Possible output3 :

```
hey from message3
hey from message1
hey from message2
```

Explenation:

Since "getMessage3" method doesn’t have any parameter/ doesn’t depend on other method, there is the possibility to print its print statement first, then the "getMessage1" method will run because it’s the same as a "getMessage3" method, and lastly the "getMessage2" method’s will print its print statement (the "getMessage1" method and the "getMessage3" method are all in the same level now, that’s why there were 2 possible outputs).

Question4:

-Possible output1 :

```
hey from message1
hey from Main controller
hey from message3
hey from message2
```

Explenation:

Since the "getMessage1" method doesn’t have any parameter/ doesn’t depend on other method, its print statement will print first, and then the print statement in "MainController" constructor will run due to the presence of a qualifier ("1"); this constructor can’t run before the "getMessage1" method and depend on its parameter. "getMessage3" method will run the third one because it’s the same as the "getMessage1" method, then the "getMessage2" method will be the last one since this method depends on the "getMessage3" method and can’t run before it.

****

-Possible output2 :

```
hey from message1
hey from message3
hey from message2
hey from Main controller
```

Explenation:

In this case, the "getMessage1" method will run first, then "getMessage2" after it, since these two methods are at the same level and don’t have any parameters or depend on other methods. After that, the "getMessage2" method will run, which is this method depends on the "getMessage3" method due to the presence of a qualifier ("3"), so it will not run before it. The last one is the "MainController" constructor, which also depends on the "getMessage1" method.

****

-Possible output3 :

```
hey from message1
hey from message3
hey from Main controller
hey from message2
```

Explenation:

In this case, again, "getMessage1" and "getMessage3" methods will run first because they don’t have any parameters and don’t depend on other methods. "MainController" constructor and "getMessage2" now in the same level (the constructor depends on the "getMessage1" method and the method depends on the "getMessage3" method); the difference between this possible output and the last one is simply that the "MainController" constructor will run before the "getMessage2" method.

****
-Possible output4 :

```
hey from message3
hey from message2
hey from message1
hey from Main controller
```

Explenation:

Since the "getMessage3" method is in the same level as the "getMessage1" method, which is all these methods don’t have any parameters or don’t depend on other methods, the difference now and as another possibility that could happen is that the "getMessage3" method will run first, and after that, the "getMessage2" method will run since this method depends on the "getMessage3" method, then the "getMessag1" method will run, and the last one is the "MainController" constructor, which depends on that method.

****
-Possible output5 :

```
hey from message3
hey from message1
hey from message2
hey from Main controller
```

Explenation:

In this case, again, "getMessage1" and "getMessage3" methods will run first because they don’t have any parameters and don’t depend on other methods. The "getMessage2" method can’t run until the "getMessage3" method runs first, due to the presence of a qualifier ("3") that makes the "getMessage2" method depend on the "getMessage3" method. The last one is the "MainController" constructor, which is the same thing but depends on the "getMessage1" method.

****
-Possible output6 :

```
hey from message3
hey from message1
hey from Main controller
hey from message2
```

Explenation:

In this case, again, "getMessage1" and "getMessage3" methods will run first because they don’t have any parameters and don’t depend on other methods. The difference in this case from the previous case is simply that the "getMessage2" method and the "MainController" constructor are now in the same leve, so, in this case, which is one possibility that could happen, the "MainController" constructor will run before the "getMessage2" method.

Question5:

-Possible output :

```
hey from message3
hey from message2
hey from Main controller
hey from message1
```

Explenation:

In this case, since the "getMessage3" method is the only one that doesn’t depend on any method and doesn’t have a parameter, it will run first. The "getMessage2" method must run directly after the "getMessage3" method because the "getMessage2" method depends on it, and no other method depends on the "getMessage3" method. After that, the "MainController" constructor will run directly after the "getMessage2" method because it’s dependent on that method due to the presence of a qualifier ("2"). The last one that will run is the "getMessage1" method since it depends on the "MainController" constructor.
