Q1 :

@SpringBootApplication  
public class SpringPollApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringPollApplication.class, args);  
 }  
  
  
 @Bean  
 @Qualifier("1")  
 public String getMessage1() {  
 System.*out*.println("hey from message1");  
 return "1";  
 }  
}

hey from message1

**Explanation :**

one possible result due to have one function only.

Q2 :

@SpringBootApplication  
public class SpringPollApplication {  
  
 public static void main(String[] args) {SpringApplication.*run*(SpringPollApplication.class, args);}  
  
 @Bean  
 @Qualifier("1")  
 public String getMessage1() {  
 System.*out*.println("hey from message1");  
 return "1";  
 }  
  
 @Bean  
 public String getMessage2(@Qualifier("1") String data) {  
 System.*out*.println("hey from message2");  
 return "data";  
 }  
}

hey from message1  
 hey from message2

**Explanation :**

one result only due to have Qualifier in function (getMessage2) which make function (getMessage1) always implement first.

Q3 :

@SpringBootApplication  
public class SpringPollApplication {  
  
 public static void main(String[] args) {SpringApplication.*run*(SpringPollApplication.class, args);}  
  
 @Bean  
 @Qualifier("1")  
 public String getMessage1(){  
 System.*out*.println("hey from message1");  
 return "1";  
 }  
  
 @Bean  
 @Qualifier("2")  
 public String getMessage2(@Qualifier("3") String data){  
 System.*out*.println("hey from message2");  
 return "data" ;  
 }  
  
 @Bean  
 @Qualifier("3")  
 public String getMessage3(){  
 System.*out*.println("hey from message3");  
 return "3" ;  
 }  
}

hey from message1  
 hey from message3  
 hey from message2

or

hey from message3  
 hey from message2  
 hey from message1

or

hey from message3  
 hey from message1  
 hey from message2

**Explanation :**

since function (getMessage2) Qualified by (getMessage3) then won’t be implemented until (getMessage3) implemented so the arrangement will be based on appear of (getMessage1) or (getMessage3) , note if (getMessage3) implemented first it could implement (getMessage2) second because it is has been Qualified.

Q4

@SpringBootApplication  
public class SpringPollApplication {  
  
 public static void main(String[] args) {SpringApplication.*run*(SpringPollApplication.class, args);}  
  
 @Bean  
 @Qualifier("1")  
 public String getMessage1(){  
 System.*out*.println("hey from message1");  
 return "1";  
 }  
  
 @Bean  
 @Qualifier("2")  
 public String getMessage2(@Qualifier("3") String data){  
 System.*out*.println("hey from message2");  
 return "data" ;  
 }  
  
 @Bean  
 @Qualifier("3")  
 public String getMessage3(){  
 System.*out*.println("hey from message3");  
 return "3" ;  
 }  
}

@Component  
public class MainController {  
  
 String data ;  
  
 public MainController(@Qualifier("1") String data){  
 this.data = data ;  
 System.*out*.println("hey from Main controller");  
 }  
}

hey from message1

hey from Main controller

hey from message3

hey from message2

or

hey from message1

hey from message3

hey from Main controller

hey from message2

or

hey from message1

hey from message3

hey from message2

hey from Main controller

or

hey from message3

hey from message1

hey from message2

hey from Main controller

or

hey from message3

hey from message2

hey from message1

hey from Main controller

or

hey from message3  
 hey from message1

hey from Main controller

hey from message2

**Explanation :**

Functions (getMessage1) and (getMessage3) don’t need to be Qualified so propabbly one of them implemented before the other and based on the implemented function (getMessage2) or the (Constructor) will be Qualified and be among the odds of being second or third or forth, note this is for the function that has been Qualified but in other side the other function will implemented lastly or thirdly in case function Qualified by it implemented secondly.

Q5

@SpringBootApplication  
public class SpringPollApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(SpringPollApplication.class, args);  
 }  
  
 @Bean  
 @Qualifier("1")  
 public String getMessage1(MainController mainController){  
 System.*out*.println("hey from message1");  
 return "1";  
 }  
  
 @Bean  
 @Qualifier("2")  
 public String getMessage2(@Qualifier("3") String data){  
 System.*out*.println("hey from message2");  
 return "data" ;  
 }  
  
 @Bean  
 @Qualifier("3")  
 public String getMessage3(){  
 System.*out*.println("hey from message3");  
 return "3" ;  
 }  
}

@Component  
public class MainController {  
  
 String data ;  
  
 public MainController(@Qualifier("2") String data){  
 this.data = data ;  
 System.*out*.println("hey from Main controller");  
 }  
}

hey from message3

hey from message2

hey from Main controller

hey from message1

**Explanation :**

since function (getMessage1) have Constructor argument and won’t be implemented until the Constructor implemented and the Constructor Qualified by (getMessage2) which Qualified by (getMessage3) so always (getMessage3) will implemented first then (getMessage2) then the Constructor then (getMessage1)