Describe Spring AOP?  
What is the difference between concern and cross-cutting concern in Spring AOP?  
What are the available AOP implementations?  
What are the different advice types in spring?  
What is Spring AOP Proxy?  
What is Introduction?  
What is joint point and Point cut?  
What is Weaving in AOP?

What is the concept of AOP? Which problem does it solve?  
What is a pointcut, a join point, an advice, an aspect, weaving, Introduction, Target Object, AOP Proxy?  
How does Spring solve (implement) a cross cutting concern?  
Which are the limitations of the two proxy-types?  
How many advice types does Spring support. What are they used for?  
What do you have to do to enable the detection of the @Aspect annotation?  
Name three typical cross cutting concerns?  
What two problems arise if you don’t solve a cross cutting concern via AOP?  
What does @EnableAspectJAutoProxy do?  
What is a named pointcut?  
How do you externalize pointcuts? What is the advantage of doing this?  
What is the JoinPoint argument used for?  
What is a ProceedingJoinPoint?  
What are the five advice types called?  
Which advice do you have to use if you would like to try and catch exceptions?  
Limitations of Spring AOP?  
What are the supported AspectJ pointcut designators in Spring AOP?  
How to declare aspect in Spring AOP?  
How to declare a pointcut in Spring AOP?  
What do you understand by Load-time weaving (LTW) in Spring?