**1> Explain Spring AOP**

Aspect-oriented programming (AOP) refers to a programming technique that allows programmers to modularize crosscutting concerns or behavior that cuts across the typical divisions of responsibility, such as logging, security and transaction management.

**2> what is Aspect in AOP ?**

It is a module which has a set of APIs providing cross-cutting requirements. For example, a logging module would be called AOP aspect for logging. An application can have any number of aspects depending on the requirement. In spring AOP, aspects are implemented using regular classes annotated with @Aspect annotation

**3 > What is the difference between concern and cross-cutting concern in Spring AOP?**

The concern is behavior that we want to have in a particular module of an application. A concern may be defined as a functionality we want to implement.

The cross-cutting concern is a concern which is applicable throughout the application and it affects the entire application. For example, logging, security and data transfer is applicable for every module of an application hence they are cross-cutting concerns.

**4> What is Join point ?**

The Join point represents a point in an application where we can plug-in an AOP aspect. It is the actual place in the application where an action will be taken using Spring AOP framework.

**5> what is Advice ? Explain its Types.**

Advice represents an action taken by an aspect at a particular join point.

This is the actual piece of code that is invoked during the program execution by the Spring AOP framework.

There are five types of advice in spring AOP.

I. before: Run advice before the a method execution.

II. after: Run advice after the a method execution regardless of its outcome.

III. after-returning: Run advice after the a method execution only if method completes successfully.

IV. after-throwing: Run advice after the a method execution only if method exits by throwing an exception.

V. around: Run advice before and after the advised method is invoked.

**6> What is pointcut ?**

The pointcut is a set of one or more joinpoints where an advice should be executed. You can specify pointcuts using expressions or patterns.

**7> What is Introduction in Spring AOP?**

An Introduction allows us to add new methods or attributes to existing classes.

**8> What is Target object in Spring AOP?**

The target object is an object being advised by one or more aspects. It will always be a proxy object.

It is also referred to as the advised object.

**9> What is a Proxy in Spring AOP?**

A proxy is an object that is created after applying advice to a target object. when you think of client objects the target object and the proxy object are the same.

**10> What are the different types of AutoProxying in Spring AOP?**

I. BeanNameAutoProxyCreator,

II. DefaultAdvisorAutoProxyCreator,

III. Metadata autoproxying.

**11> What is weaving in Spring AOP? What are the different points where weaving can be applied?**

Weaving is the process of linking aspects with other application types or objects to create an advised object.

Weaving can be done at compile time, at load time, or at runtime.

**12> How spring AOP works internally?**

Spring AOP is proxy-based. Spring uses either JDK proxies (preferred wheneven the proxied target implements at least one interface) or CGLIB proxies (if the target object does not implement any interfaces) to create the proxy for a given target bean.

Unless configured to do otherwise, Spring AOP performs run-time weaving. You can however set up Spring to do load-time weaving through AspectJ.