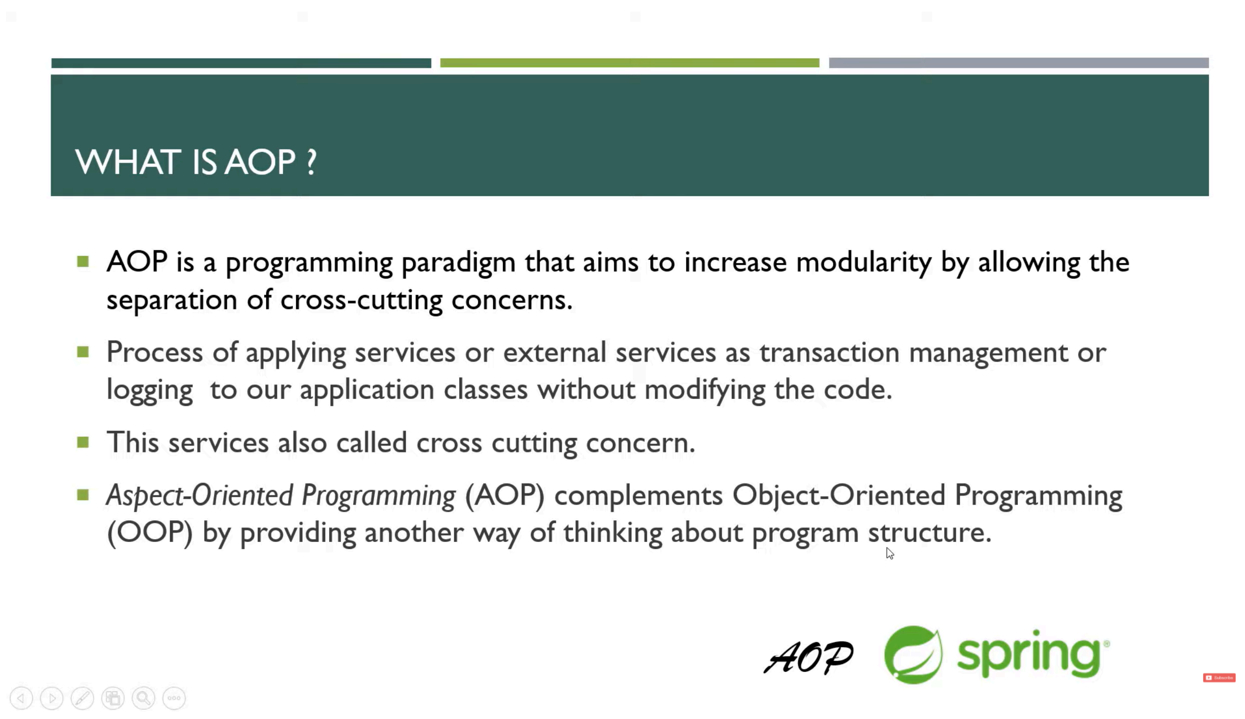
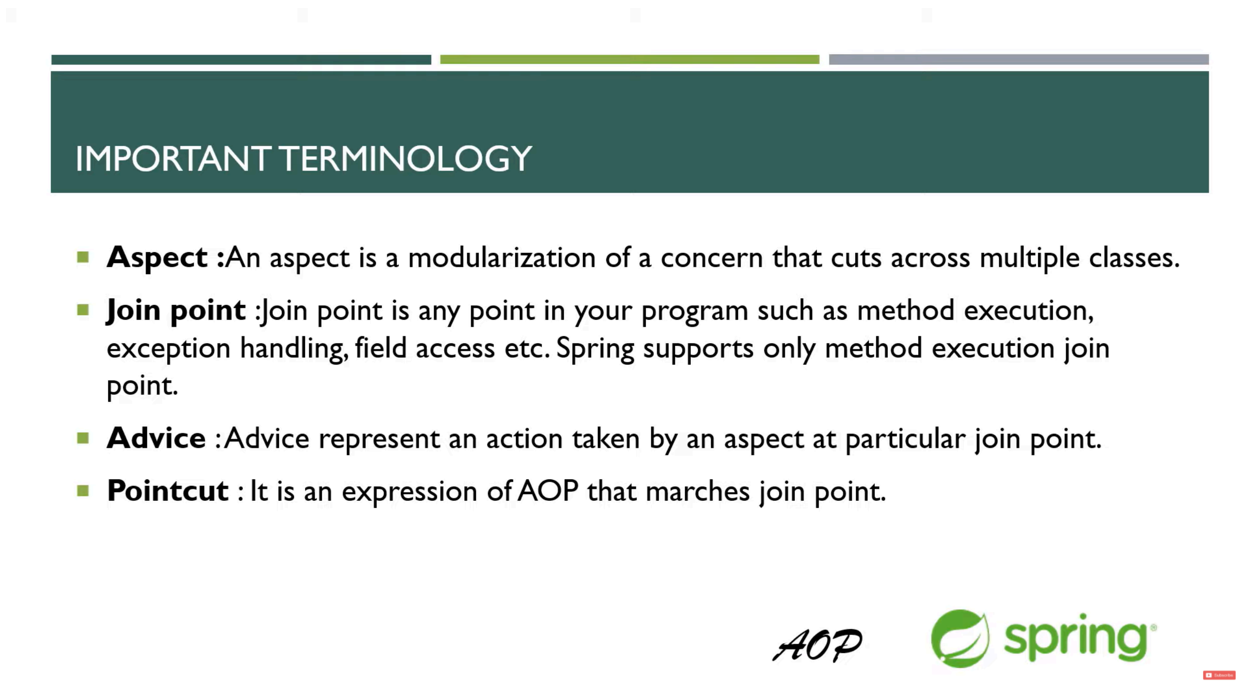


WHAT IS AOP ?

- AOP is a programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns.
- Process of applying services or external services as transaction management or logging to our application classes without modifying the code.
- This services also called cross cutting concern.
- *Aspect-Oriented Programming* (AOP) complements Object-Oriented Programming (OOP) by providing another way of thinking about program structure.

AOP spring



IMPORTANT TERMINOLOGY

- **Aspect :**An aspect is a modularization of a concern that cuts across multiple classes.
- **Join point** :Join point is any point in your program such as method execution, exception handling, field access etc. Spring supports only method execution join point.
- **Advice** :Advice represent an action taken by an aspect at particular join point.
- **Pointcut** : It is an expression of AOP that marches join point.

AOP spring

Mainly used for:
a) Logging
b) security
c) transactions management

Aspect: a class that has cross cutting concerns

Join point: a point during execution (method call, constructor call, etc)
Advice: the action taken at a join point (logging before method)
Pointcut: Expression that selects join points (which method to intercept). It is predicate that defines where advice should be applied.
Weaving: Linking aspects with code (at runtime using proxies)
Target object: the actual object whose method is being intercepted.

Advice: action taken by the aspect at particular join point. Five types:
1. Before: executed before method call.
2. After: executed after method call.
3. AfterReturning: executed after method returns a result, but not if an exception occurs.
4. Around: surrounds the method execution, allowing we to control the method execution and its result.
5. AfterThrowing: executed if the method throws an exception.

AOP Framework:

1. **AspectJ:** a powerful and mature AOP framework that supports compile time and load time weaving. It offers full AOP support with its own syntax and tools.
2. **JBoss AOP:** part of JBoss application server, offering integration with Java EE applications.
3. **Spring AOP:** a simpler proxy based framework that integrates with spring framework, using XML configurations or annotations to define aspect and pointcuts.

For enable AOP in spring:
Using annotation: @EnableAspectJAutoProxy
Using xml: <aop:aspectj-autoproxy />

Dependecies to include:

1. Spring core
2. Spring context
3. Spring AOP
4. AspectJRT
5. AspectJweaver
   Make sure that in aspectjrt and aspectjweaver dependency, we don’t include scope in dependency, it is runtime so we want to remove it.