Spring AOP Tutorial

1. [What is AOP?](https://www.javatpoint.com/spring-aop-tutorial)
2. [Where use AOP?](https://www.javatpoint.com/spring-aop-tutorial)
3. [AOP Concepts and Terminology](https://www.javatpoint.com/spring-aop-tutorial)
4. [Spring AOP Implementation](https://www.javatpoint.com/spring-aop-tutorial)

**Aspect Oriented Programming** (AOP) compliments OOPs in the sense that it also provides modularity. But the key unit of modularity is aspect than class.

AOP breaks the program logic into distinct parts (called concerns). It is used to increase modularity by **cross-cutting concerns**.

A **cross-cutting concern** is a concern that can affect the whole application and should be centralized in one location in code as possible, such as transaction management, authentication, logging, security etc.

#### Why use AOP?

It provides the pluggable way to dynamically add the additional concern before, after or around the actual logic.

### AOP Implementations

AOP implementations are provided by:

1. AspectJ
2. Spring AOP
3. JBoss AOP