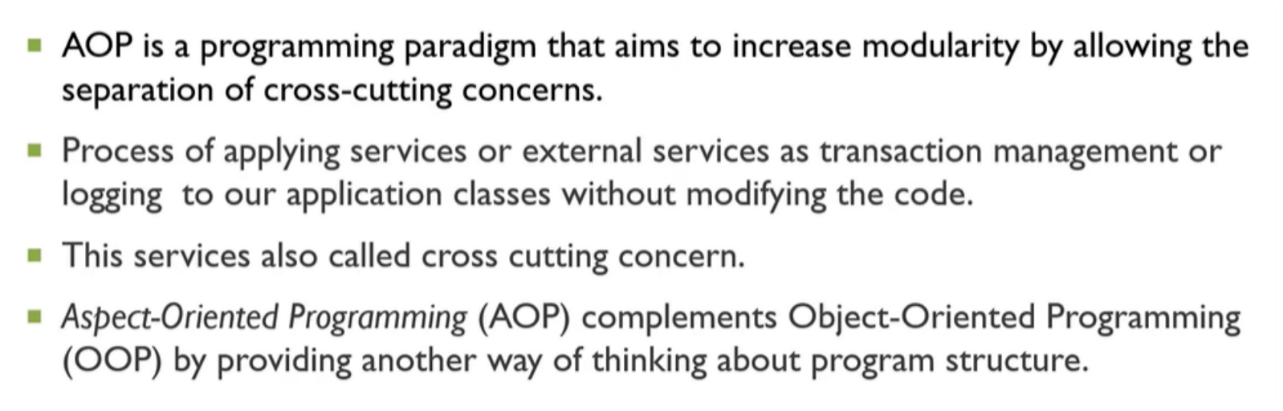
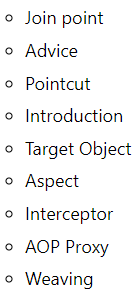
**What is AOP?**



**What are all terminology in AOP?**



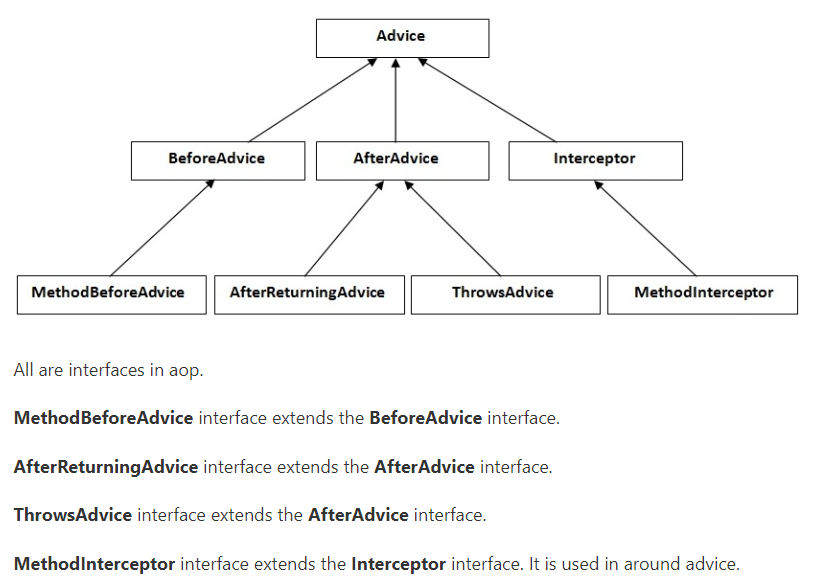
**What all dependency required for AOP?**



**What does it mean all?**

* **Before Advice**: it executes before a join point.
* **After Returning Advice**: it executes after a joint point completes normally.
* **After Throwing Advice**: it executes if method exits by throwing an exception.
* **After (finally) Advice**: it executes after a join point regardless of join point exit whether normally or exceptional return.
* **Around Advice**: It executes before and after a join point.
* **Aspect:** It is programming paradigm used for cross cutting concerns.
* **JoinPoint:** It is any point in your program such as method execution, exception handling, field access etc.
* **Pointcut:** It is an expression language of AOP that matches join points and empty method for advice.

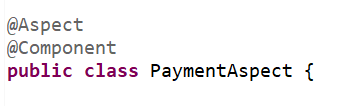
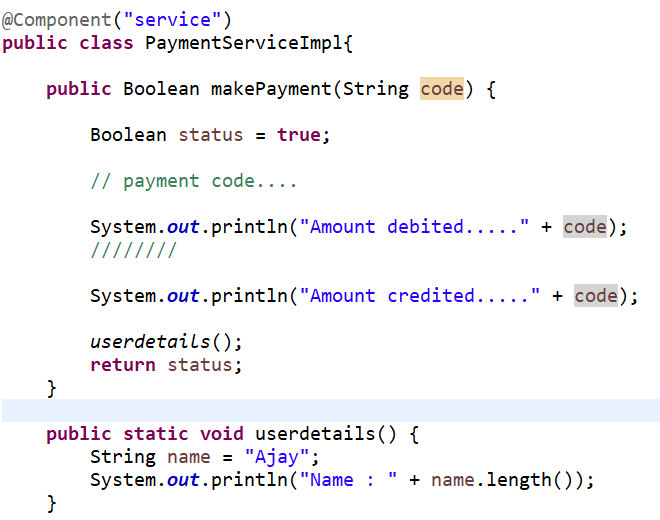
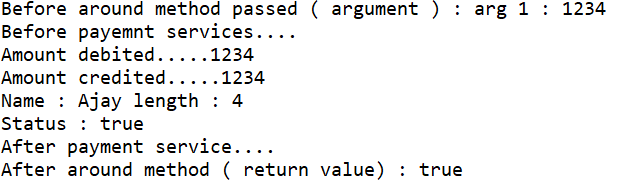
**What is hierarchy of Advice?**



**How to achieve AOP by xml and annotation?**





**How to enable Aspect in Spring?**

* By using <aop:aspect-autoproxy>
* By using @EnableAspectJAutoProxy

**What is need of AOP?**

