AspectJ is annotation variant of AOP. There will be a Business Logic Bean and AspectJ Bean. AspectJ is annotated with PointCut, Advisor and Advice. Load Business Logic Bean, spring scans and keeps Advice that are associated with Business Logic Bean and everything happens behind the scene.

Spring AOP supports 4 types of advice

* Before advice – Run before the method execution
* After returning advice – Run after the method returns a result
* After throwing advice – Run after the method throws an exception
* Around advice – Run around the method execution, combine all three advices above.

**Spring AOP without PointCut…**

1. Business Logic Bean
2. Advice or Interceptor Bean- (for aroundAdvice…. Before… Business… Around… put in try catch for throw advice)
3. A proxy (Bean) configuration which accepts Business Logic Bean and Interceptor Bean
4. Load the proxy Bean instead of Business Logic Bean.

If a class has 10 methods. If we don’t use Pointcut, then all 10 methods of a class are intercepted automatically. If we want to intercept only one or two methods, then ‘Pointcut’ comes for rescue.

* Pointcut – Indicate which method should be intercept, by method name or regular expression pattern.
* Advisor – Group ‘Advice’ and ‘Pointcut’ into a single unit, and pass it to a proxy factory object.

**Spring AOP with PointCut**

1. Business Logic Bean
2. Advice or Interceptor Bean
3. PointCut Bean (which mentions method name in RegExp or methodName itself)
4. AdvisorBean (which connects Advice and PointCut)
5. A proxy (Bean) configuration which accepts Business Logic Bean and AdvisorBean
6. Load the proxy Bean instead of Business Logic Bean.