Aspect-Oriented Programming Using Spring AOP

Contents

[1. AOP 5](#_Toc22556083)

[1.1 Benfits of AOP 6](#_Toc22556084)

[1.2 AOP Terminology 7](#_Toc22556085)

[1.3 Without AOP 8](#_Toc22556086)

[1.3.1 Car Class 8](#_Toc22556087)

[1.3.2 Car Bean declaration in AppContext.xml 8](#_Toc22556088)

[1.3.3 Main Class 9](#_Toc22556089)

[1.3.4 Run the Application 9](#_Toc22556090)

[1.4 Creating Aspect 10](#_Toc22556091)

[1.4.1 Car Class 10](#_Toc22556092)

[1.4.2 Create Aspect Class 10](#_Toc22556093)

[1.4.3 Create Log Bean is appcontext.xml 11](#_Toc22556094)

[1.4.4 Add the Dependency 12](#_Toc22556095)

[1.4.5 Enable AOP Name Space 12](#_Toc22556096)

[1.4.6 AOP Configuration 13](#_Toc22556097)

[1.4.7 Run the Application 14](#_Toc22556098)

[1.5 Referring Point cut Expression 14](#_Toc22556099)

[1.6 Creating Aspects Using Annotations 16](#_Toc22556100)

[1.6.1 Enable AspectJ deflation in appcontext.xml 16](#_Toc22556101)

[1.6.2 Annotation in Log Class 16](#_Toc22556102)

[1.6.3 Run the Application 17](#_Toc22556103)

[1.7 Getting Joint Point info in Aspects 17](#_Toc22556104)

[1.7.1 Car Class 17](#_Toc22556105)

[1.7.2 Main Class 18](#_Toc22556106)

[1.7.3 Read info from Car class Using JointPoint 18](#_Toc22556107)

[1.7.4 Run the Application 19](#_Toc22556108)

[1.8 Working with Aspects Java Based Configuration 19](#_Toc22556109)

[1.8.1 Create Configuration Class 20](#_Toc22556110)

[1.8.2 Modifying Main Class 20](#_Toc22556111)

[2. Working with Advice 21](#_Toc22556112)

[2.1 Before Advice 21](#_Toc22556113)

[2.1.1 Car Class 21](#_Toc22556114)

[2.1.2 Aspect Class 22](#_Toc22556115)

[2.1.3 Main Class 22](#_Toc22556116)

[2.1.4 Run the Application 23](#_Toc22556117)

[2.1.5 Aspect with exception during Before Advice 23](#_Toc22556118)

[2.2 Working with After Advice 24](#_Toc22556119)

[2.2.1 Main App 24](#_Toc22556120)

[2.2.2 Declaring Car Bean 25](#_Toc22556121)

[2.2.3 Run the application 25](#_Toc22556122)

[2.2.4 Create After Advice 26](#_Toc22556123)

[2.2.5 Returning value from Car Class to After Advice 28](#_Toc22556124)

[2.3 Working with After Throwing Advice 30](#_Toc22556125)

[2.3.1 Car Class 30](#_Toc22556126)

[2.3.2 Register the Car Bean in appContext.xml 31](#_Toc22556127)

[2.3.3 Main Class 31](#_Toc22556128)

[2.3.4 Create After Throwing Advice 32](#_Toc22556129)

[2.3.5 Run the Application 32](#_Toc22556130)

[2.3.6 Binding Exception in after throwing Advice 34](#_Toc22556131)

[2.4 Working with After (Finally) Advice 34](#_Toc22556132)

[2.4.1 Create After Advice 35](#_Toc22556133)

[2.4.2 Run the Application 35](#_Toc22556134)

[2.5 Working with Around Advice 36](#_Toc22556135)

[2.5.1 Car Class 37](#_Toc22556136)

[2.5.2 Main Class 37](#_Toc22556137)

[2.5.3 Registering Bean 38](#_Toc22556138)

[2.5.4 Create Around Advice 38](#_Toc22556139)

[2.5.5 Run the Application 39](#_Toc22556140)

[2.6 Working Exception with Around Advice 39](#_Toc22556141)

[2.6.1 Create Around Advice 39](#_Toc22556142)

[2.6.2 Run the Application 40](#_Toc22556143)

[2.7 Working with Return Values and Parameters 40](#_Toc22556144)

[2.7.1 Car Class 40](#_Toc22556145)

[2.7.2 Main App 41](#_Toc22556146)

[2.7.3 Create Around Advice 41](#_Toc22556147)

[2.7.4 Run the Application 42](#_Toc22556148)

[2.7.5 Override the Main Class parameter from Aspect 42](#_Toc22556149)

[2.7.6 Retuning Values from Aspect class 43](#_Toc22556150)

[2.7.7 Returning values from Car Class 45](#_Toc22556151)

[2.7.8 Run the Application 45](#_Toc22556152)

[3. Working with Point Cut 46](#_Toc22556153)

[3.1 Combining Point Cut Expression 46](#_Toc22556154)

[3.1.1 Car Class 46](#_Toc22556155)

[3.1.2 Create Point cut expression in Aspect Class 49](#_Toc22556156)

[The point cut expression will run Any method name has with dr in car class 49](#_Toc22556157)

[3.1.3 Main App 50](#_Toc22556158)

[3.1.4 Register the Bean 51](#_Toc22556159)

[3.1.5 Run the Application 51](#_Toc22556160)

[3.1.6 Create Point cut expression based on return type 52](#_Toc22556161)

[3.1.7 Run the application 53](#_Toc22556162)

[3.1.8 Combining Point Cut && operation 54](#_Toc22556163)

[3.1.9 Run the Application 54](#_Toc22556164)

[3.1.10 Combining Point Cut || operation 55](#_Toc22556165)

[3.1.11 Run the Application 55](#_Toc22556166)

[3.2 Matching Return Type Point Cut 56](#_Toc22556167)

[3.2.1 Car Class 56](#_Toc22556168)

[3.2.2 Main App 56](#_Toc22556169)

[3.2.3 No return type Point Cut Expression 57](#_Toc22556170)

[3.2.4 Run the Application 57](#_Toc22556171)

[3.2.5 Int return type Point Cut Expression 57](#_Toc22556172)

[3.2.6 Run the Application 58](#_Toc22556173)

[3.2.7 String return type Point Cut Expression 58](#_Toc22556174)

[3.2.8 Run the application 59](#_Toc22556175)

[3.2.9 Any return type 59](#_Toc22556176)

[3.2.10 Run the Application 60](#_Toc22556177)

[3.3 Matching Method Parameter Point cut Expression 60](#_Toc22556178)

[4. Spring AOP Point Cut Designators 61](#_Toc22556179)

[4.1 Matching Packages and Class Point Cut Expression 62](#_Toc22556180)

[4.1.1 Register the bean 62](#_Toc22556181)

[4.1.2 Main App 62](#_Toc22556182)

[4.1.3 Create Point Cut Expression for Any method and Any Package and Any Return Type and Any Parameter 63](#_Toc22556183)

[4.1.4 Run the application 63](#_Toc22556184)

[4.1.5 Matching the Class Point Cut Expression 64](#_Toc22556185)

[4.1.6 Run the Application 65](#_Toc22556186)

[4.1.7 Matching the Package Point Cut Expression without Sub Packages 65](#_Toc22556187)

[4.1.8 Run the Application 66](#_Toc22556188)

[4.1.9 Matching the Package Point Cut Expression with Sub Packages 66](#_Toc22556189)

[4.1.10 Run the Application 67](#_Toc22556190)

[4.2 Using Args Point Cut Expression 67](#_Toc22556191)

[4.2.1 Car Class 67](#_Toc22556192)

[4.2.2 Main Class 68](#_Toc22556193)

[4.2.3 No Parameter Args Point Cut Expression 68](#_Toc22556194)

[4.2.4 Run the Application 69](#_Toc22556195)

[4.2.5 One Parameter Point Cut Expression 69](#_Toc22556196)

[4.2.6 Run the Application 70](#_Toc22556197)

[4.2.7 Two Parameter Point Cut Expression 70](#_Toc22556198)

[4.2.8 Run the Application 71](#_Toc22556199)

[4.2.9 One Parameter (Any Type) Point Cut Expression 71](#_Toc22556200)

[4.2.10 Run the Application 72](#_Toc22556201)

[4.2.11 Any Number of Parameter Point Cut Expression 72](#_Toc22556202)

[4.2.12 Read the Parameter values in Aspect Class 73](#_Toc22556203)

[4.2.13 Run the Application 73](#_Toc22556204)

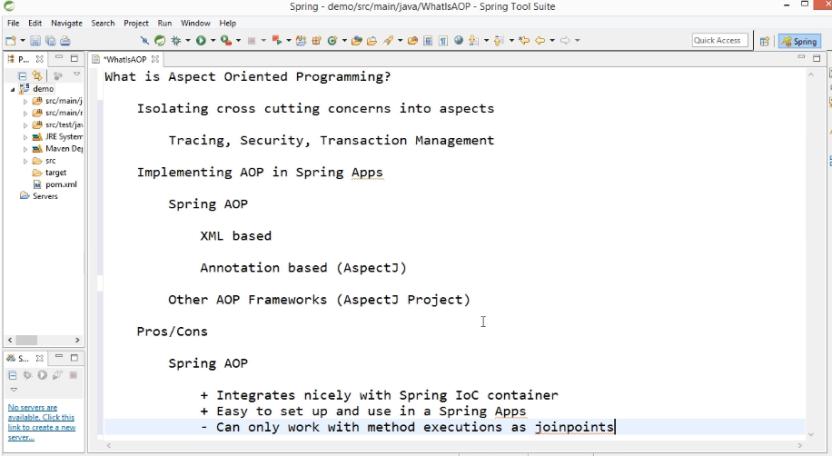
[4.3 Using Bean Point Cut Expression 74](#_Toc22556205)

[4.3.1 Main App 74](#_Toc22556206)

[4.3.2 Create Point Cut Expression 74](#_Toc22556207)

[4.3.3 Run the Application 75](#_Toc22556208)

# 1. AOP

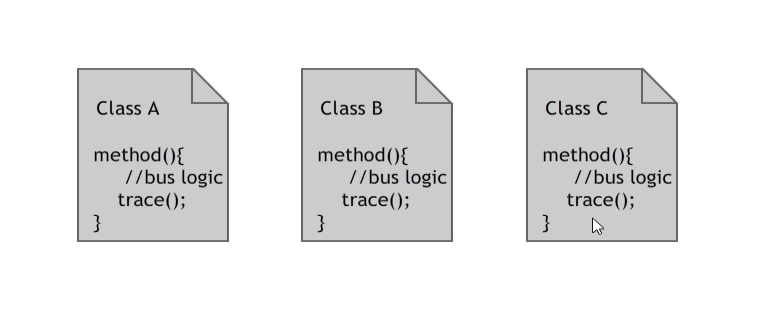




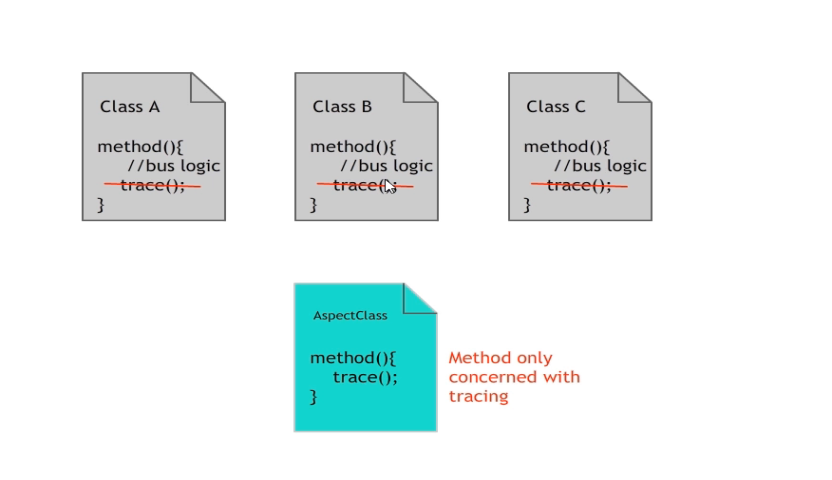
## Benfits of AOP

AOP mains use for Manage code easily. Concentrate only business logic for every class Instead of adding log file in every class. We can manage centralized logger file for access the log.

Without AOP

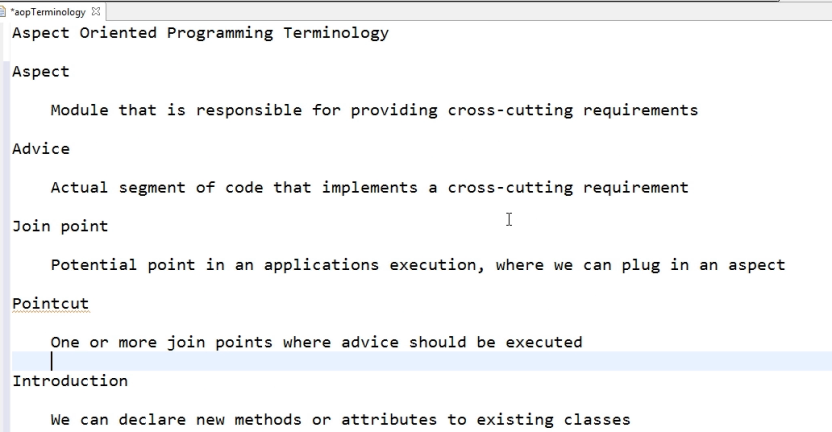


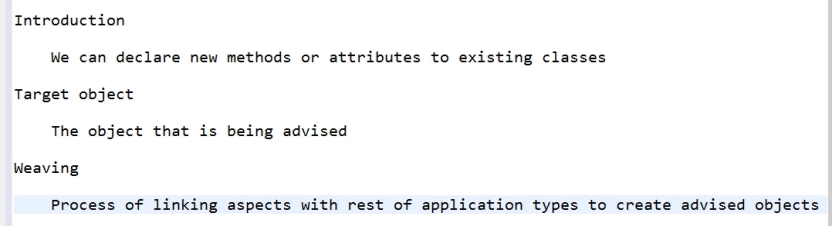
Manage only business logic using AOP



## AOP Terminology

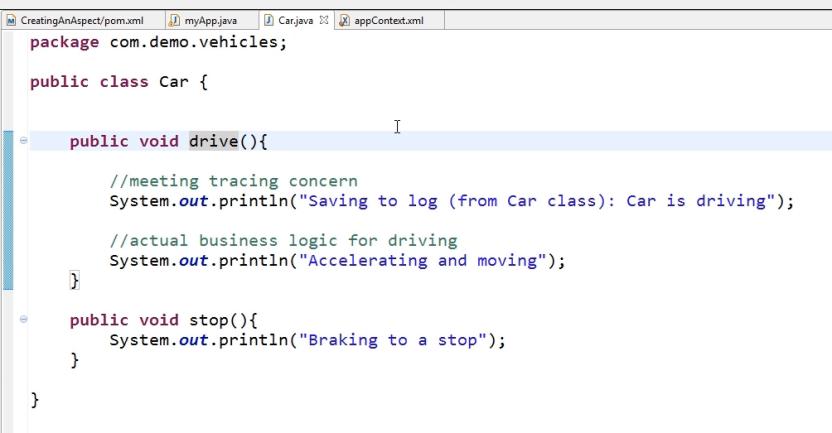
**Cross-Cutting:** same module is repeating on every class. This module is not a part of business logic.





## Without AOP

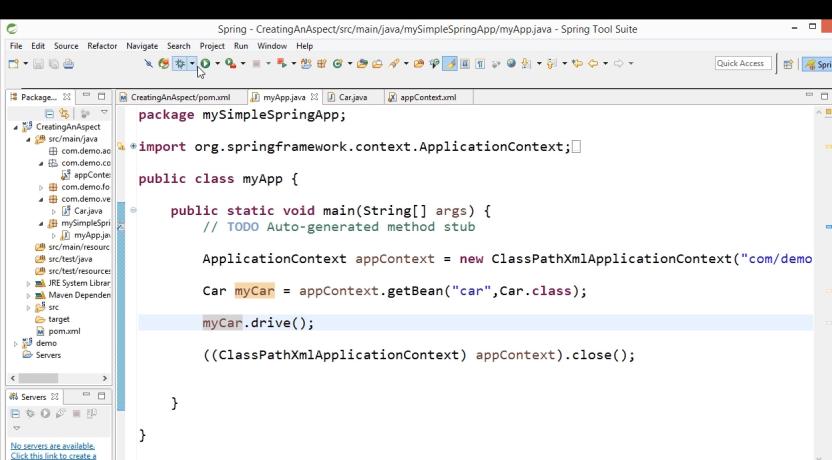
### 1.3.1 Car Class



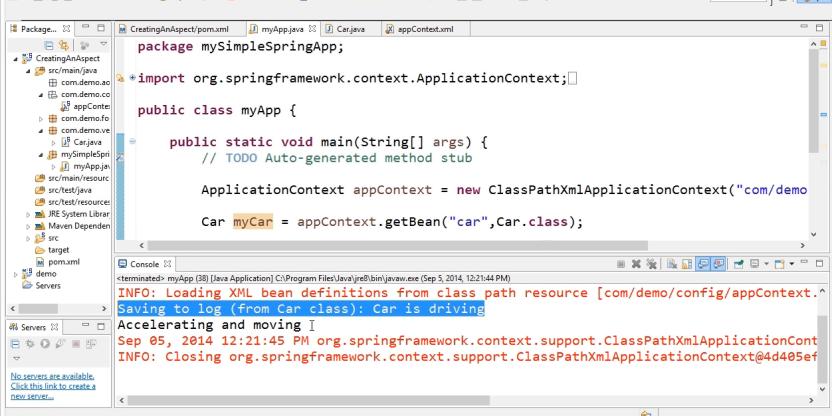
### 1.3.2 Car Bean declaration in AppContext.xml



### 1.3.3 Main Class

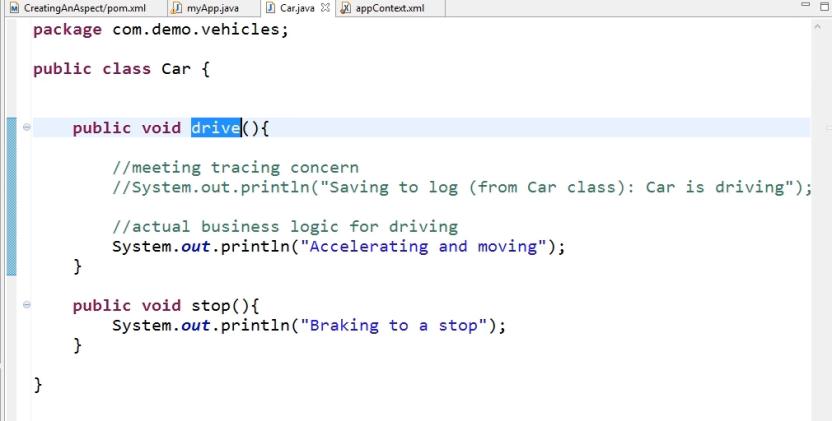


### 1.3.4 Run the Application

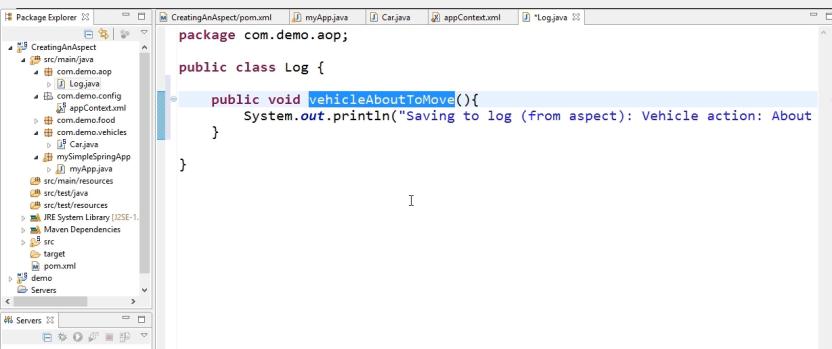


## Creating Aspect

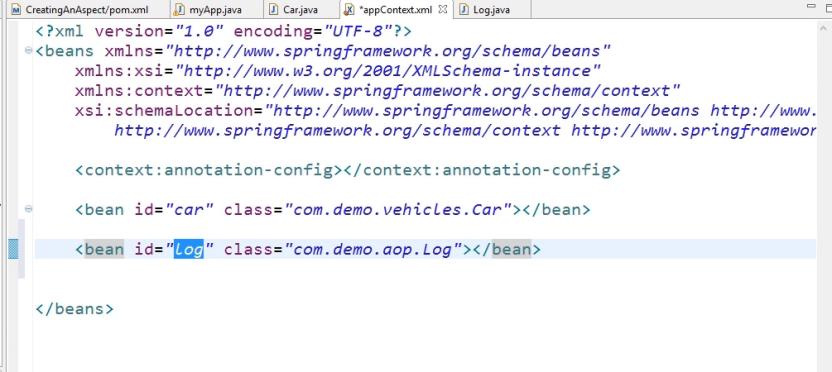
### Car Class



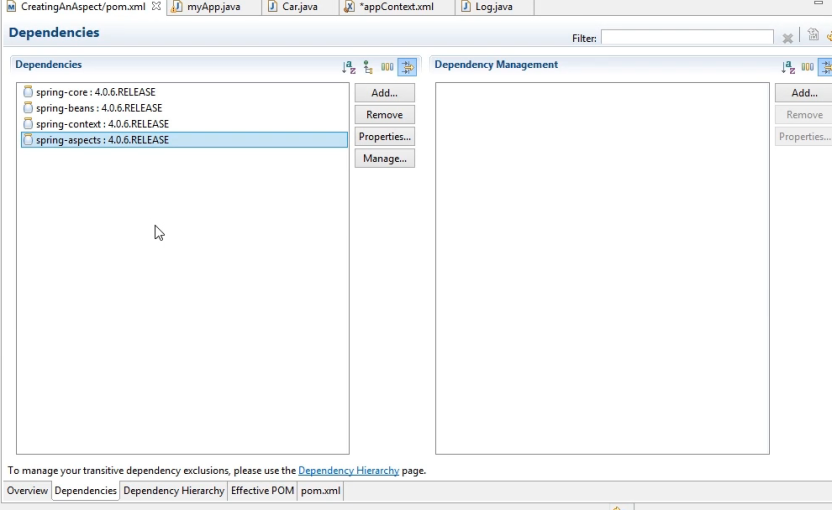
### Create Aspect Class



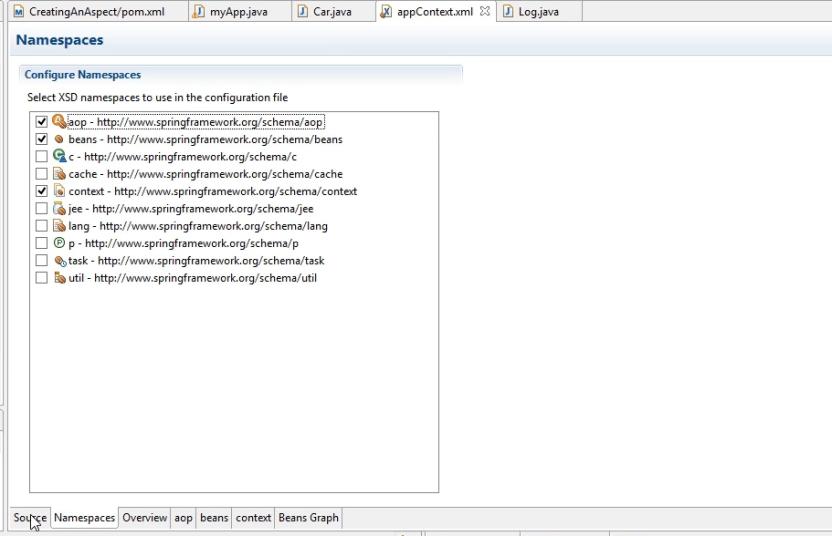
### Create Log Bean is appcontext.xml



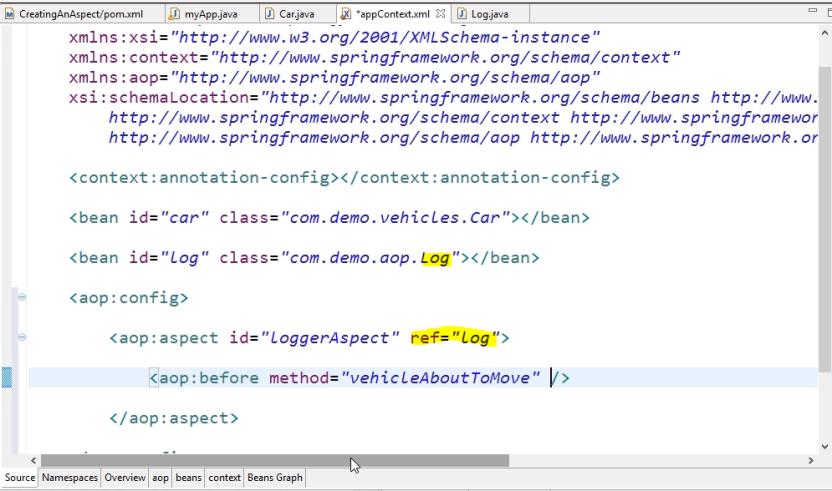
### Add the Dependency

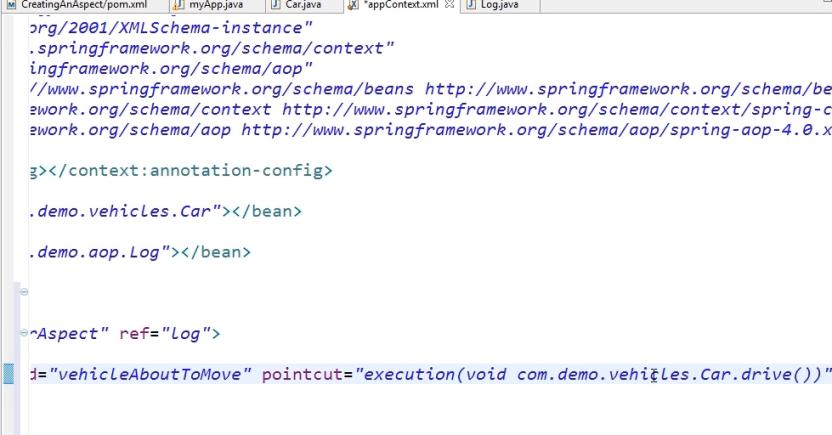


### Enable AOP Name Space

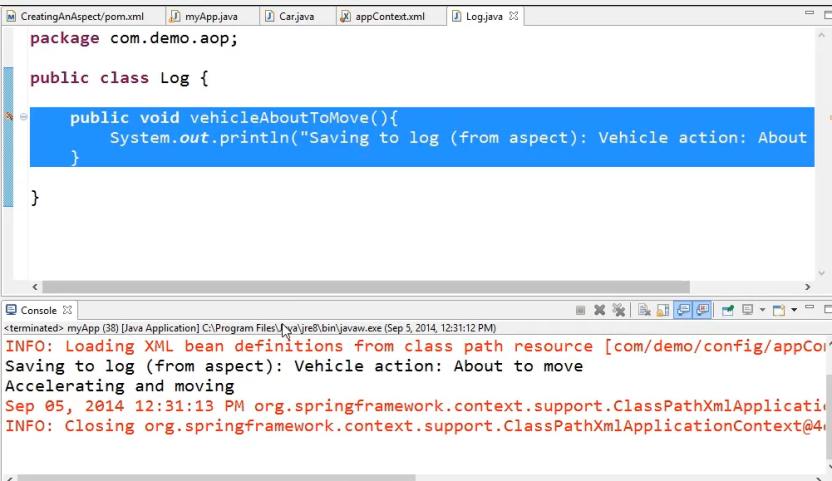


### AOP Configuration

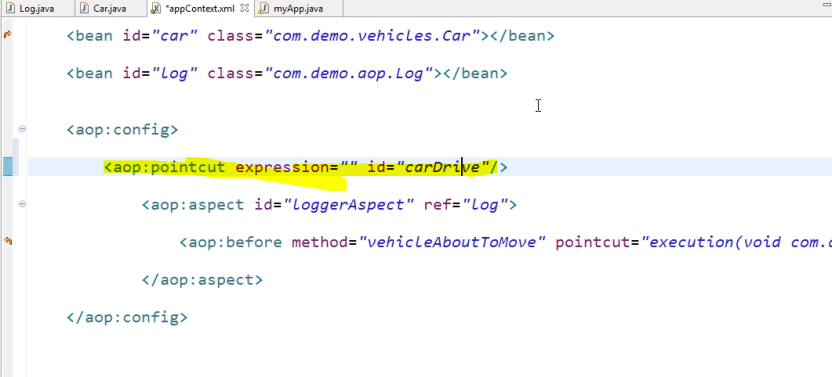


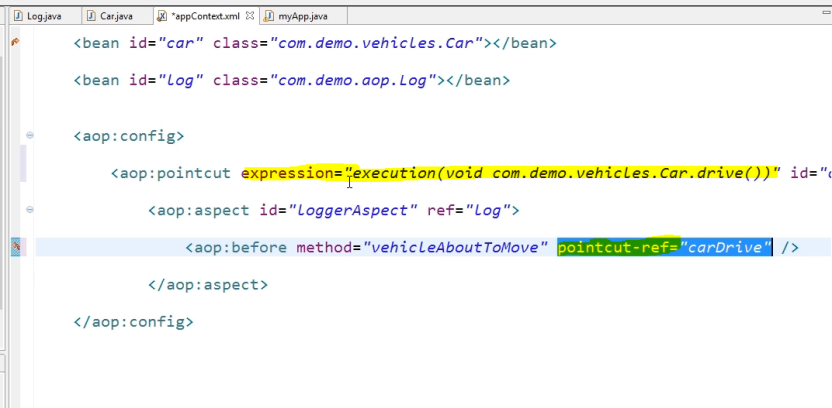


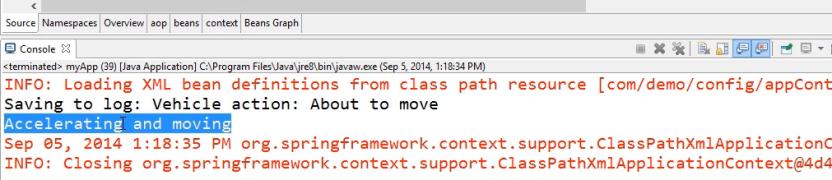
### Run the Application



## Referring Point cut Expression

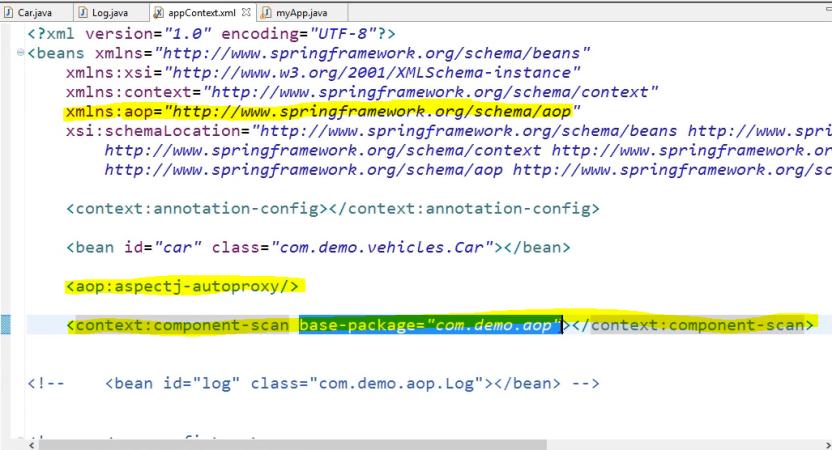




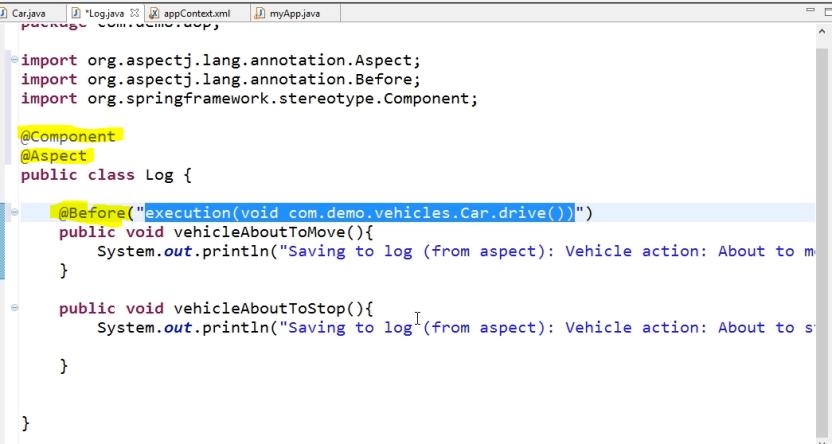


## Creating Aspects Using Annotations

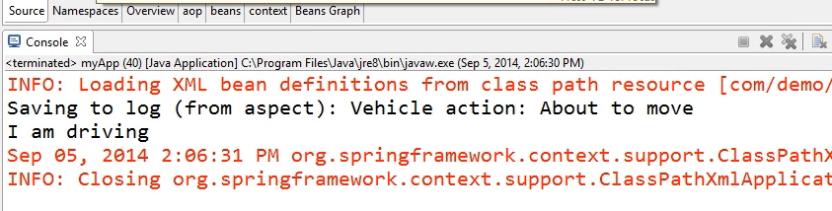
### Enable AspectJ deflation in appcontext.xml



### Annotation in Log Class

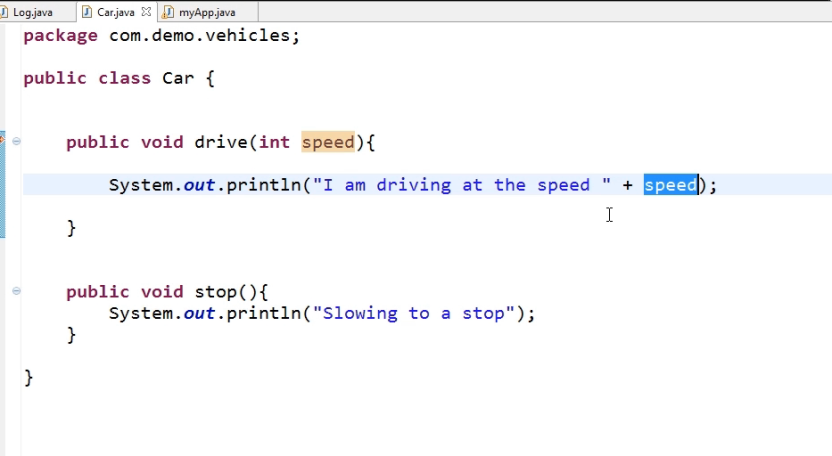


### 1.6.3 Run the Application

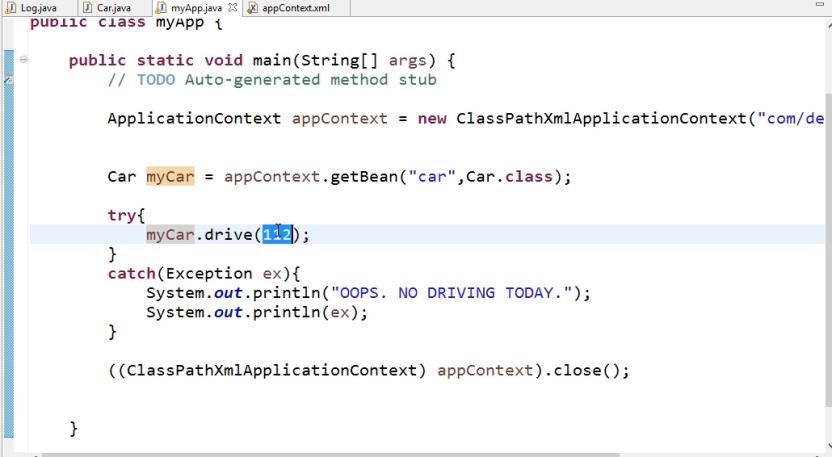


## Getting Joint Point info in Aspects

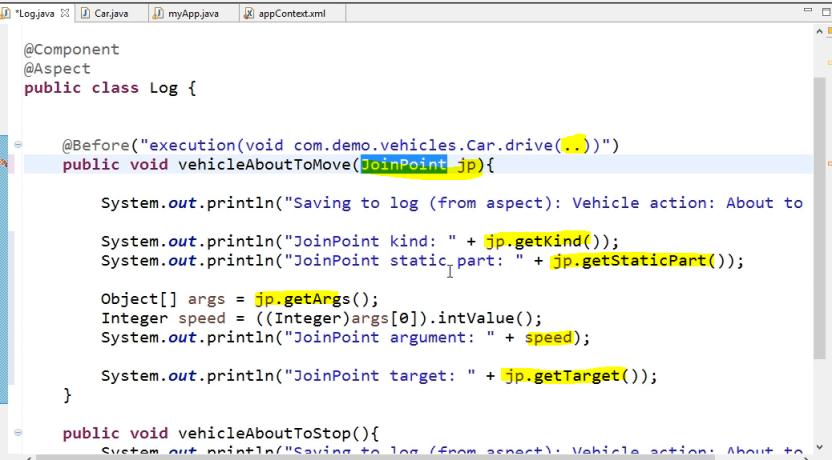
### Car Class



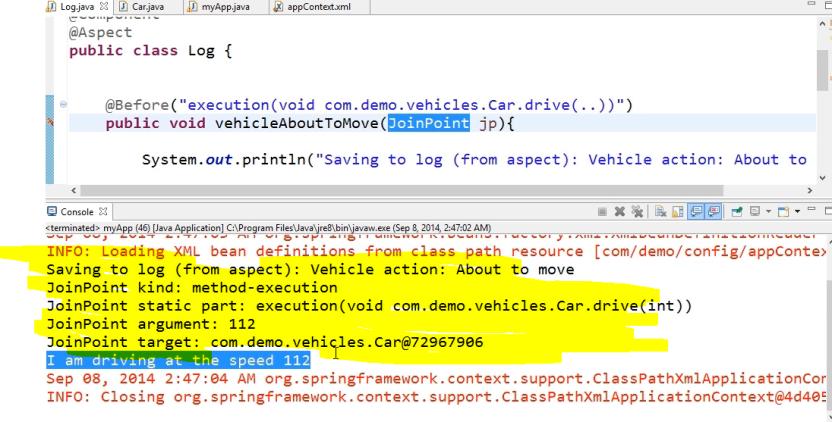
### Main Class



### Read info from Car class Using JointPoint

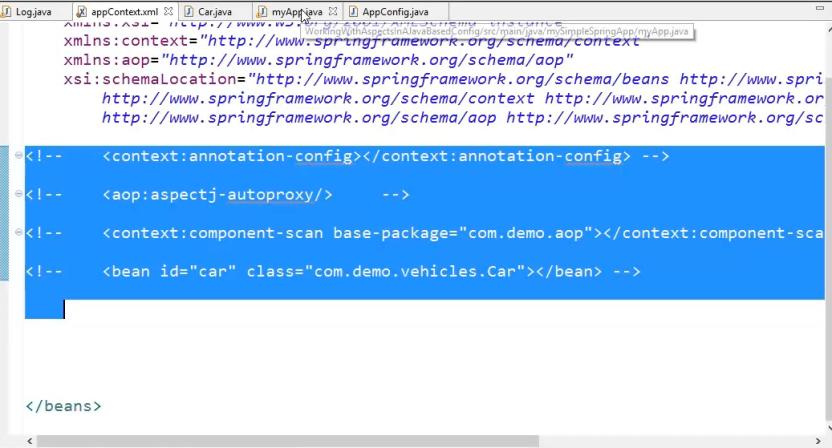


### Run the Application

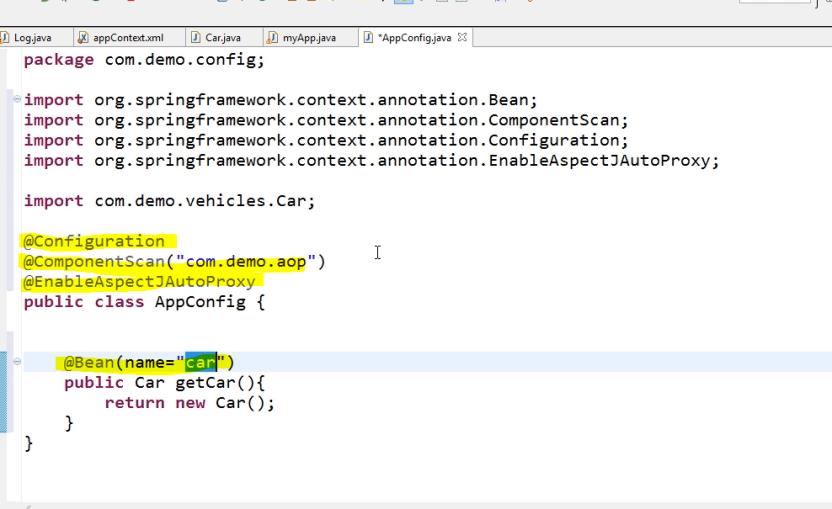


## Working with Aspects Java Based Configuration

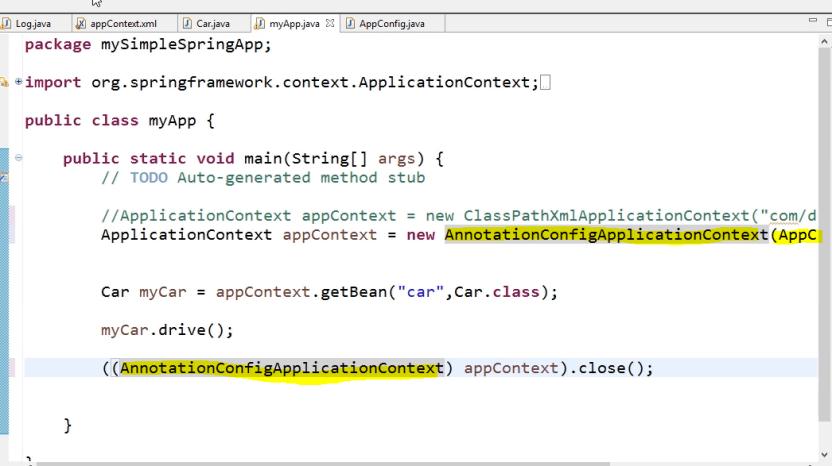
Rather than using xml configuration everything configured in Java Based Annotation



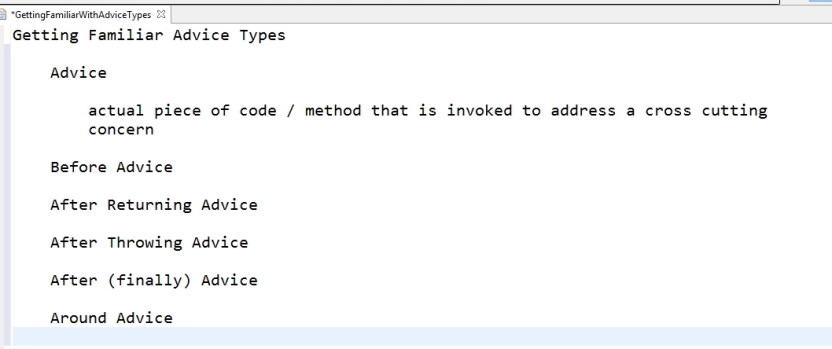
### 1.8.1 Create Configuration Class



### 1.8.2 Modifying Main Class

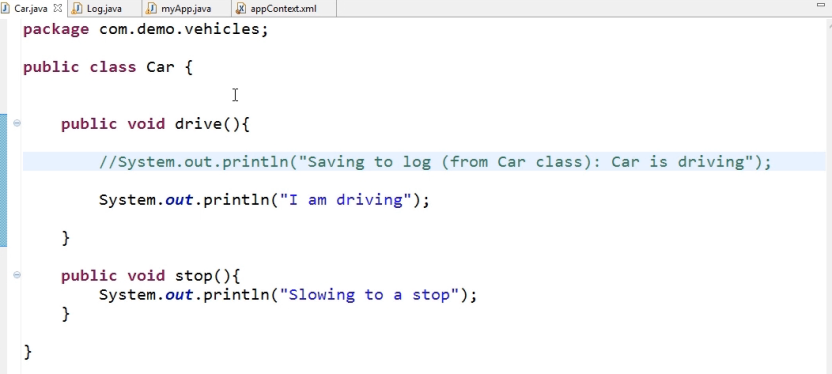


# 2. Working with Advice

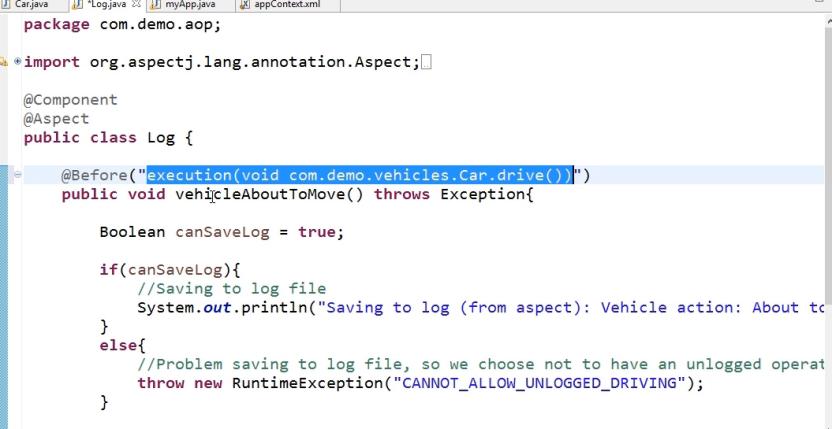


## 2.1 Before Advice

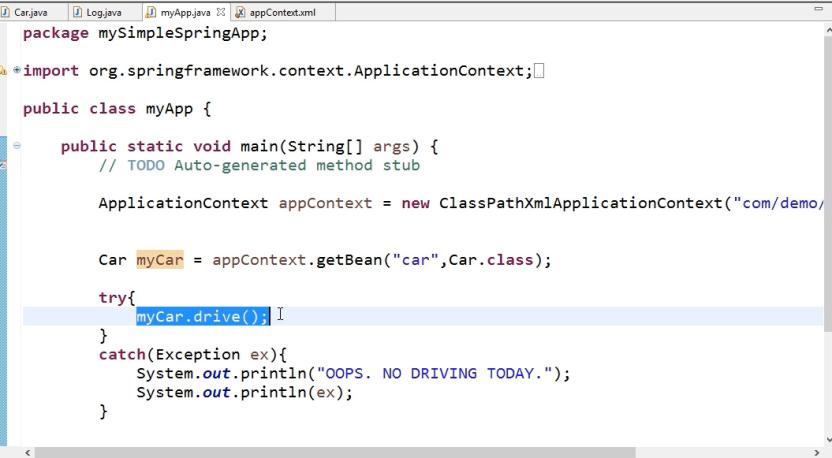
### 2.1.1 Car Class



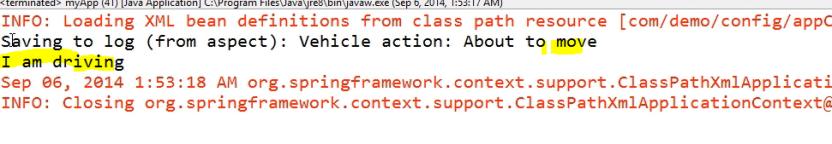
### 2.1.2 Aspect Class



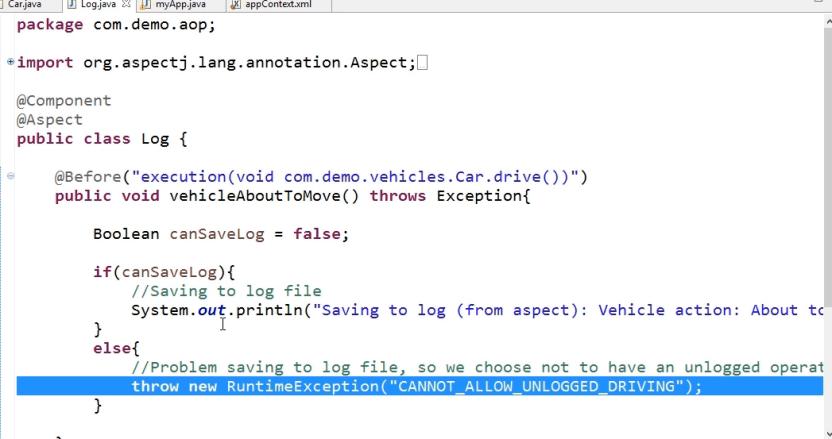
### 2.1.3 Main Class



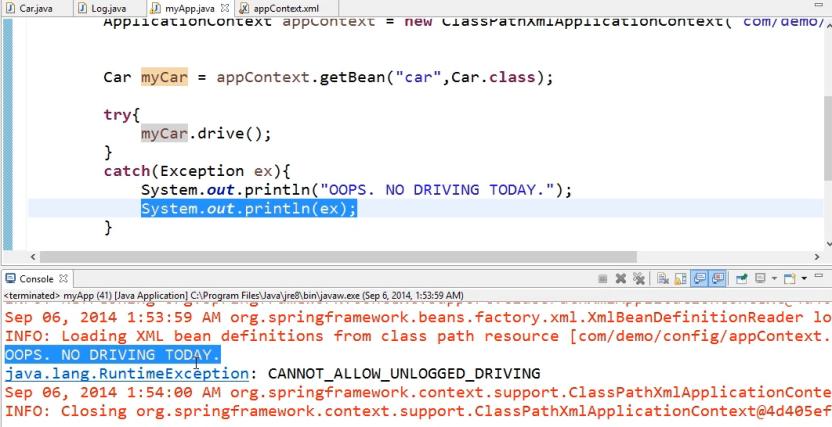
### 2.1.4 Run the Application



### 2.1.5 Aspect with exception during Before Advice

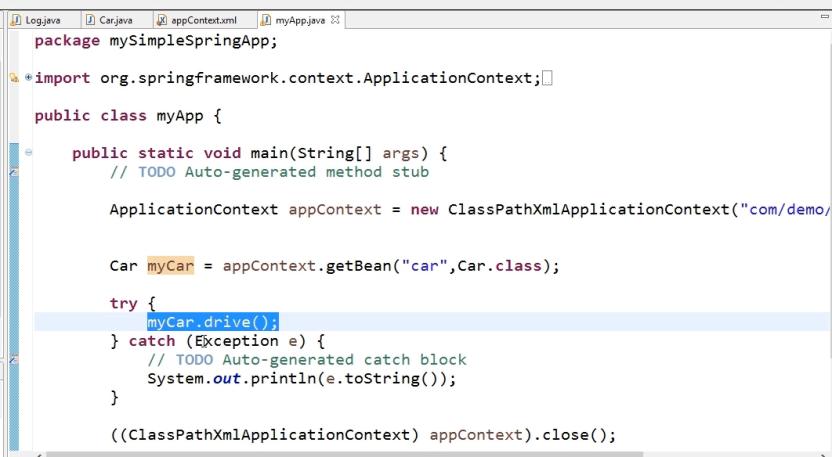


#### 2.1.5.1 Run the Application

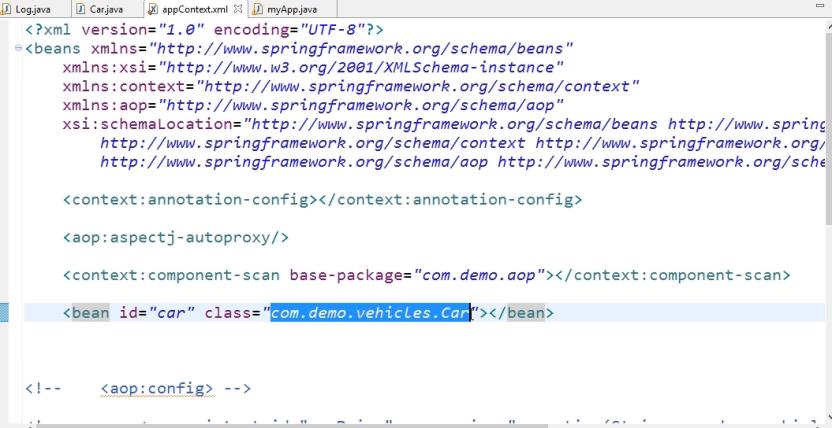


## 2.2 Working with After Advice

### 2.2.1 Main App

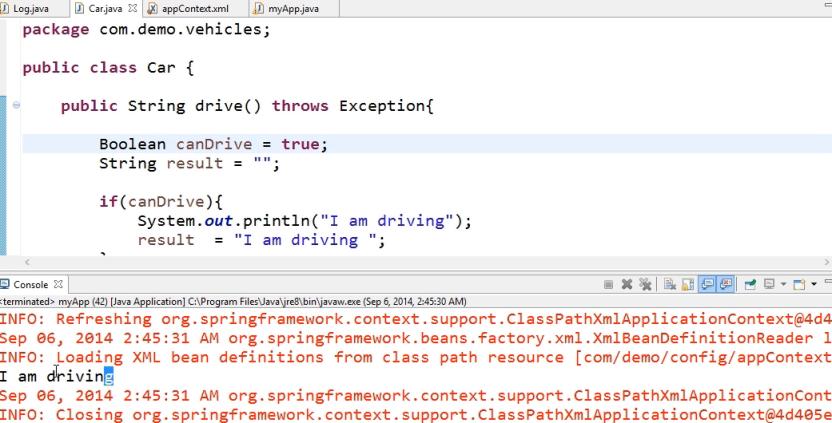


### 2.2.2 Declaring Car Bean

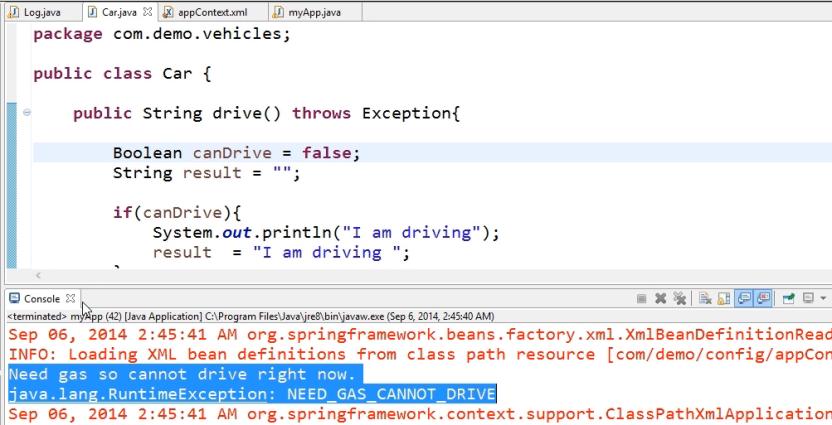


### 2.2.3 Run the application

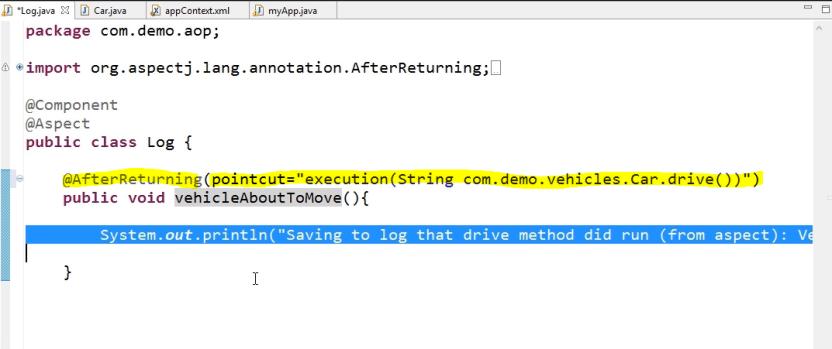
carDrive = true



carDrive=false

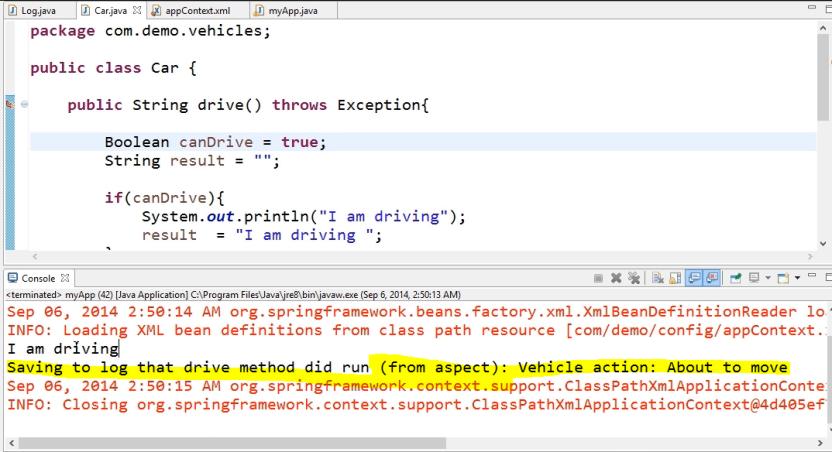


### 2.2.4 Create After Advice

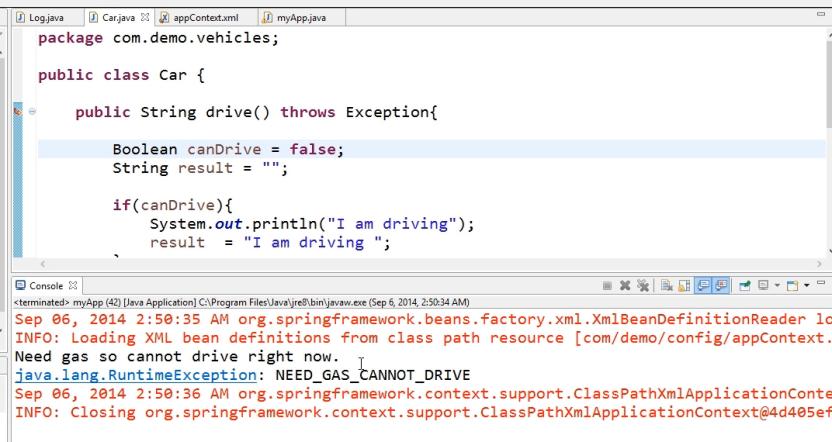


#### 2.2.4.1 Run the Application

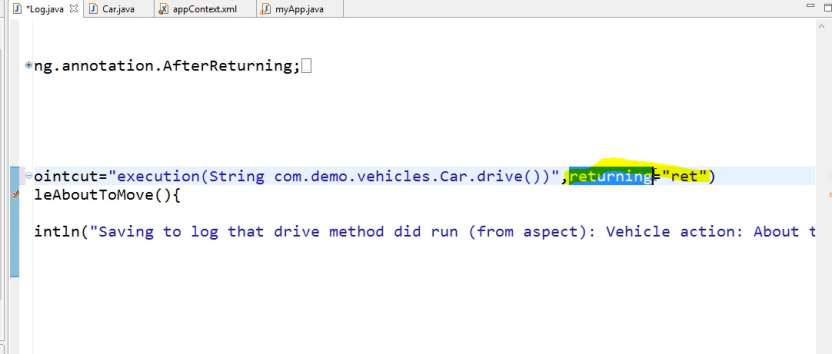
carDrive = true

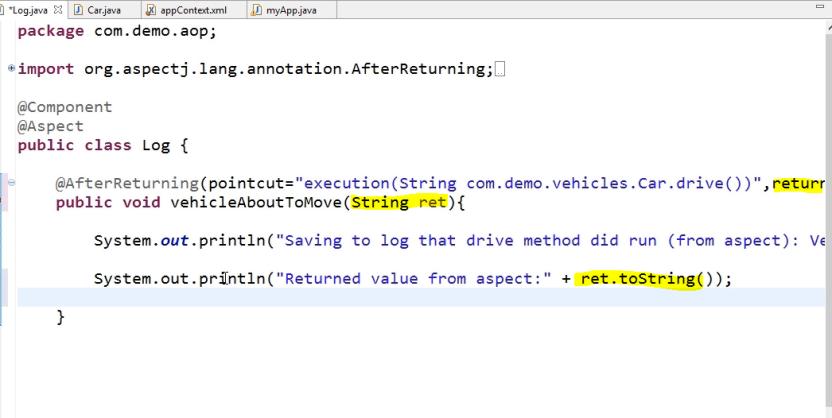


carDrive=false

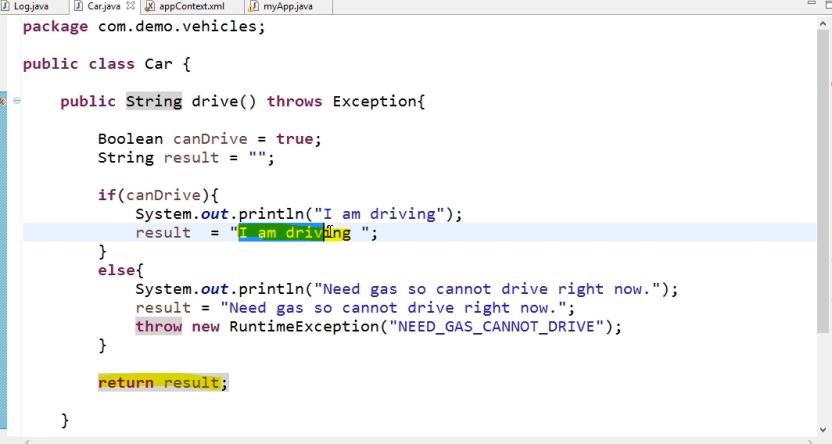


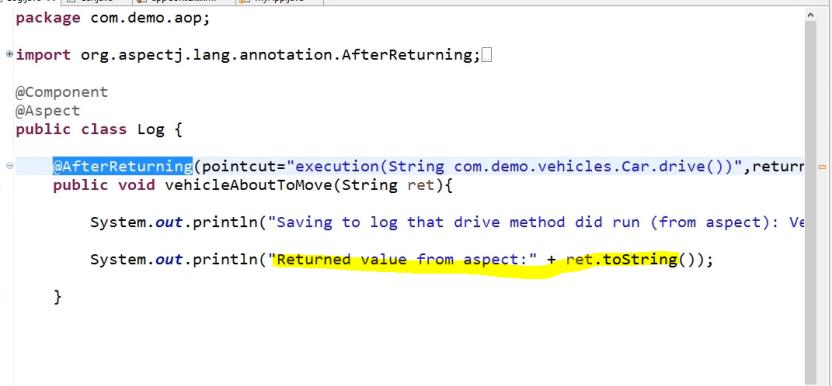
### 2.2.5 Returning value from Car Class to After Advice



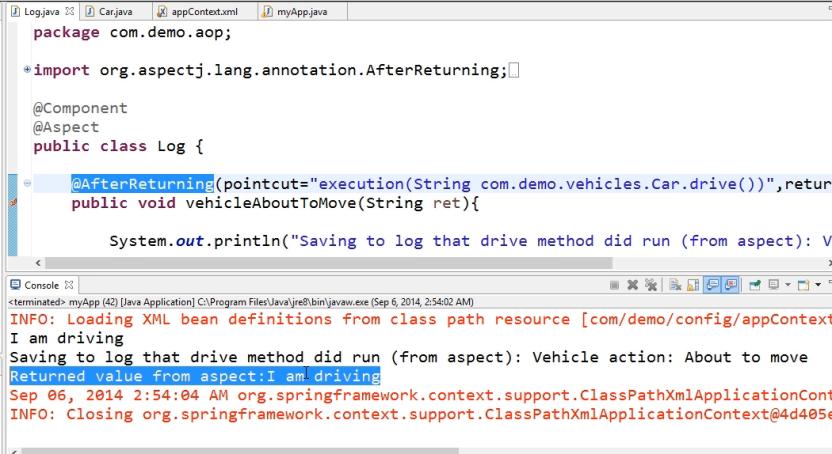


#### 2.2.5.1 Run the Application



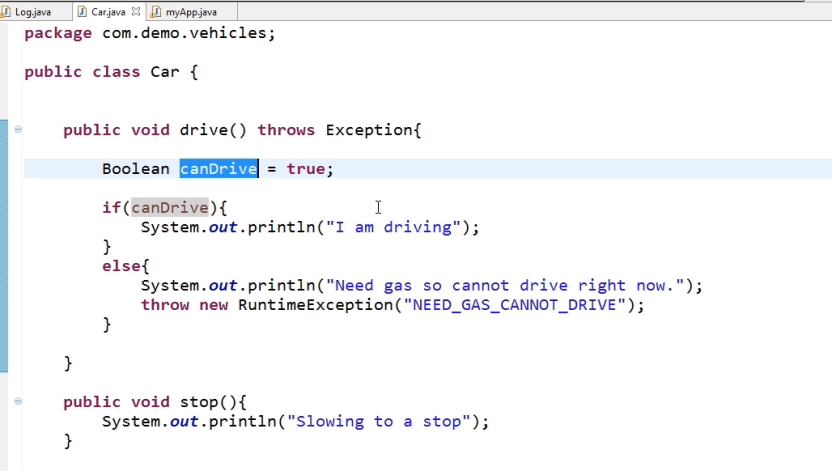


Output



## 2.3 Working with After Throwing Advice

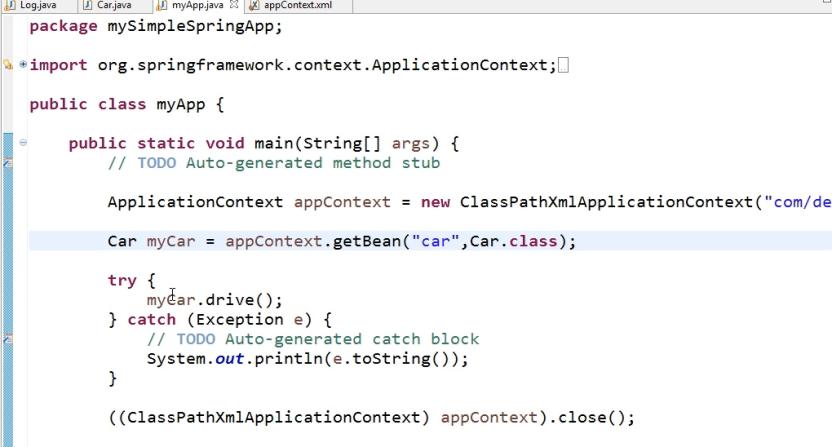
### 2.3.1 Car Class



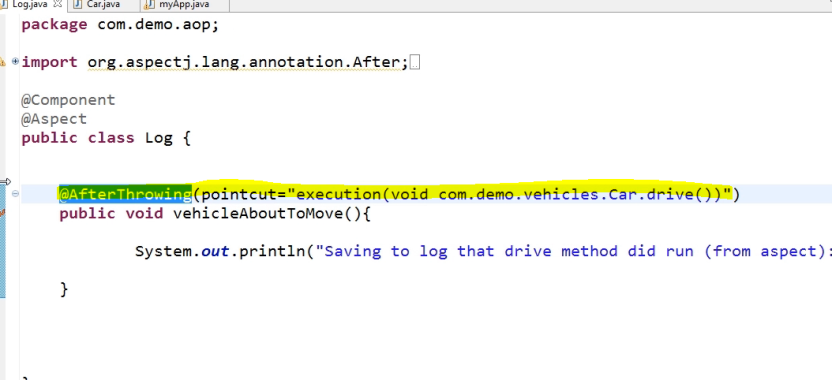
### 2.3.2 Register the Car Bean in appContext.xml



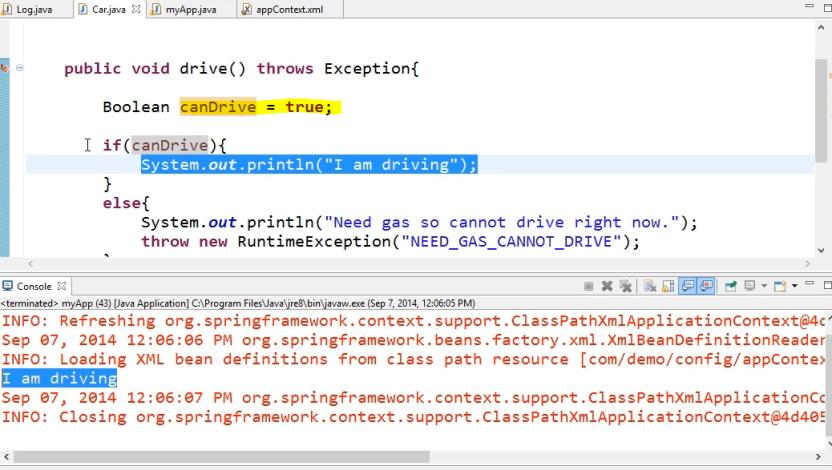
### 2.3.3 Main Class

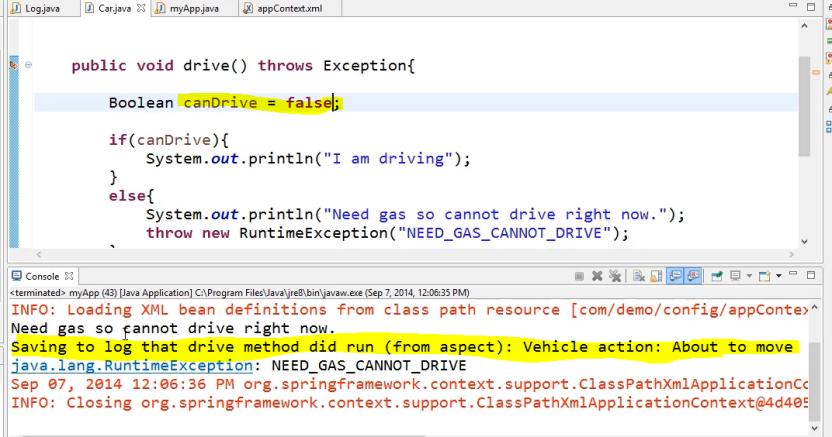


### 2.3.4 Create After Throwing Advice

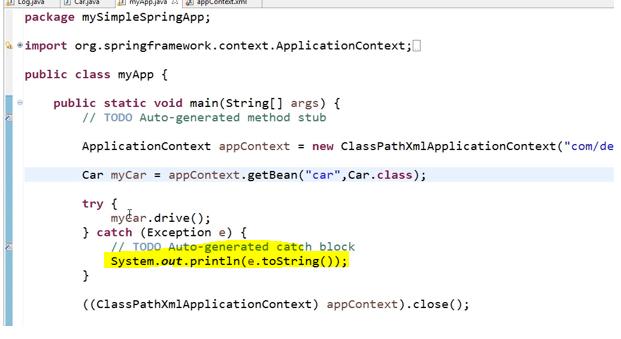


### 2.3.5 Run the Application

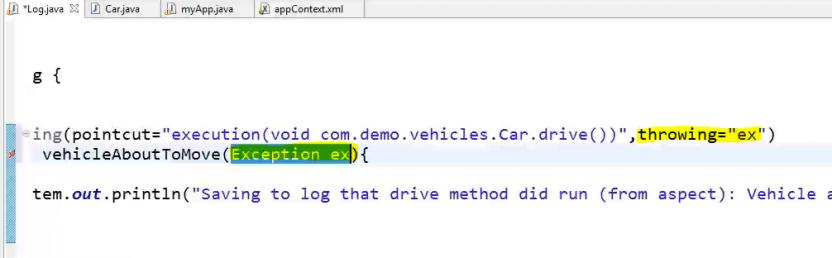


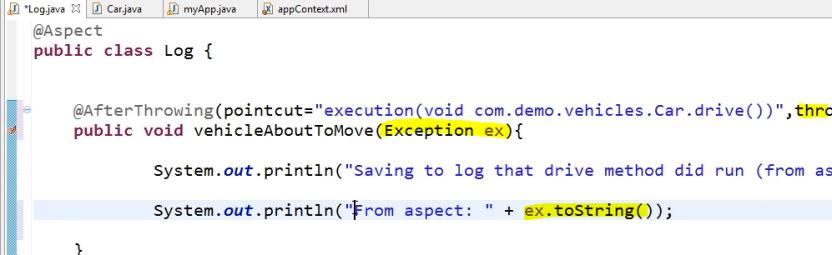


NEED\_GAS\_CANNOT\_Drive Exception name display in main app see below code

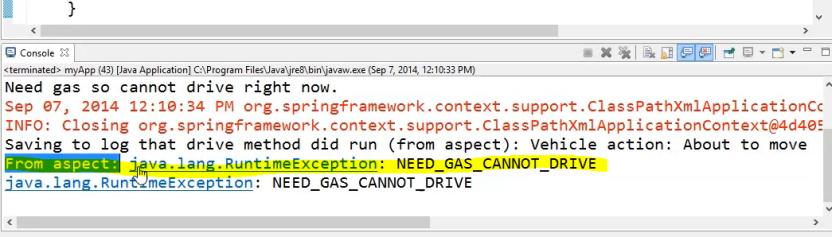


### 2.3.6 Binding Exception in after throwing Advice





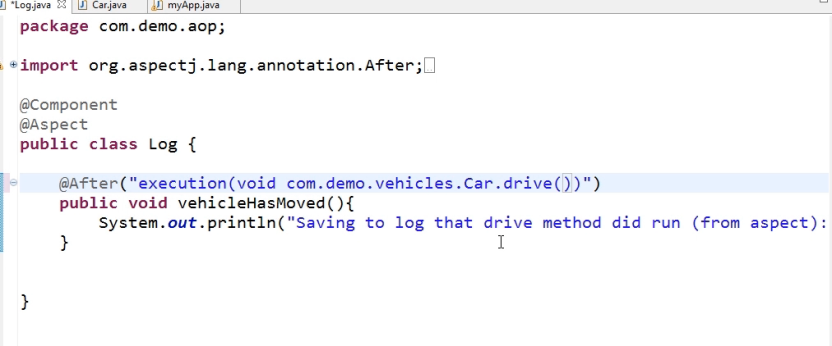
Run the Application



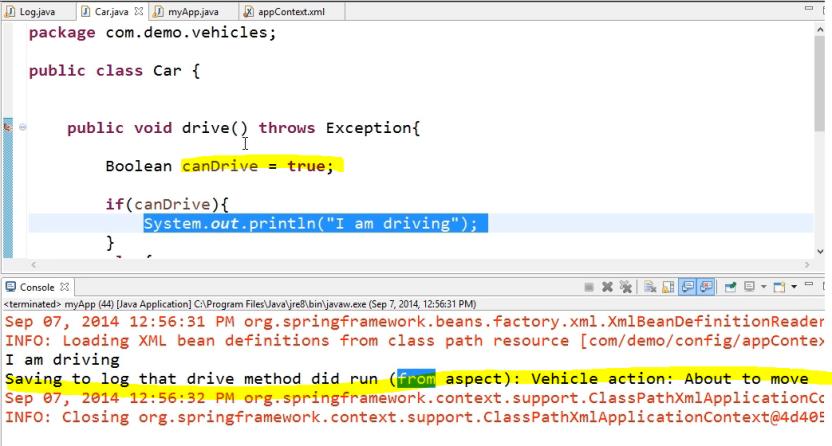
## 2.4 Working with After (Finally) Advice

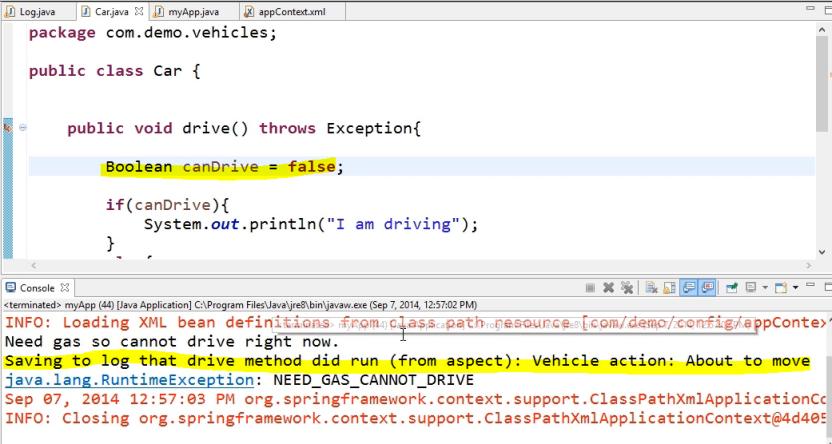
After Advice is working weather that method is run successfully or caught any exception.

### 2.4.1 Create After Advice



### 2.4.2 Run the Application





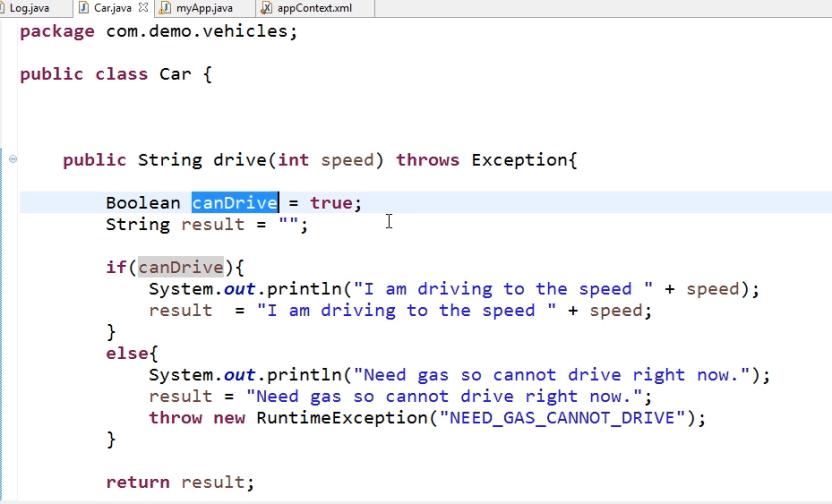
## 2.5 Working with Around Advice

Around advice will execute before method execution and after method execution.

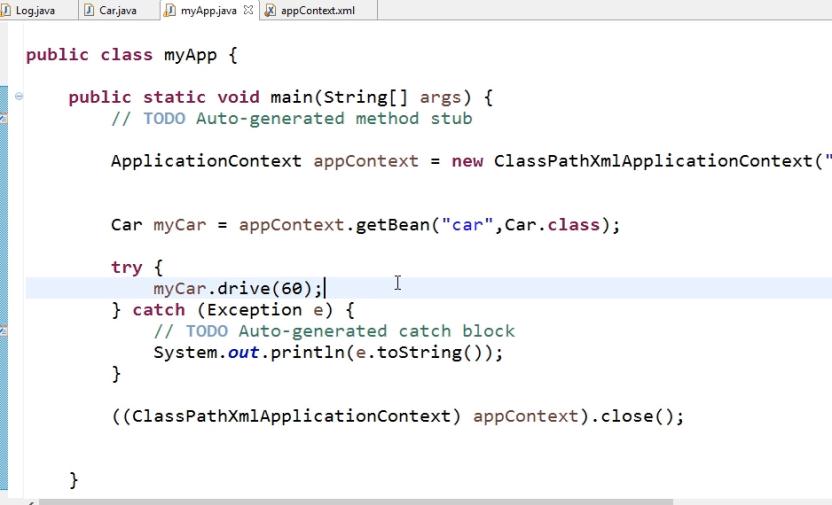
PrecendingJoinPoint : It’s used for execute certain code run before actual point cut method execution and after point cut execution.

Pjp.Procceed () should be add in Around advice. If we not added then actual point cut method won’t run.

### 2.5.1 Car Class



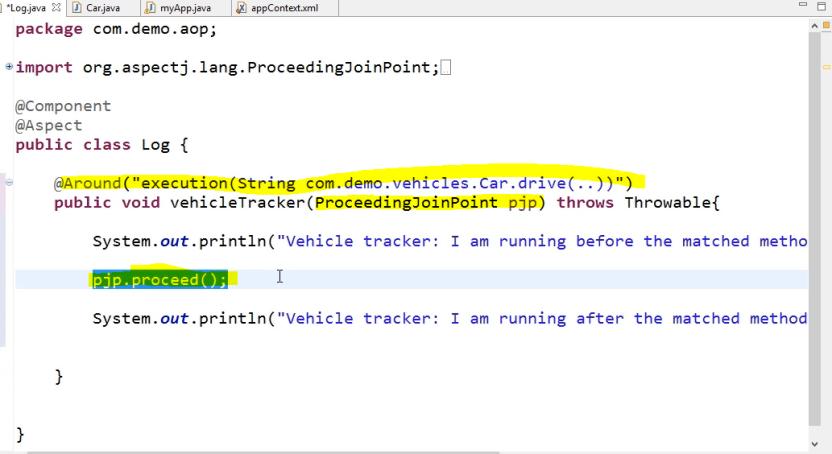
### 2.5.2 Main Class



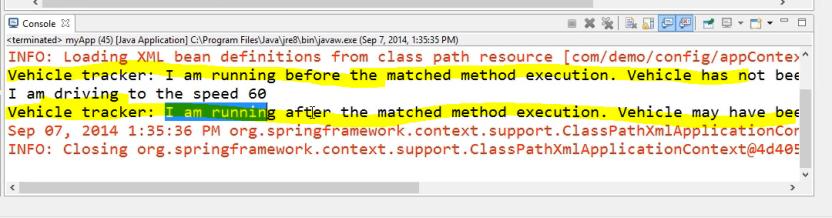
### 2.5.3 Registering Bean



### 2.5.4 Create Around Advice

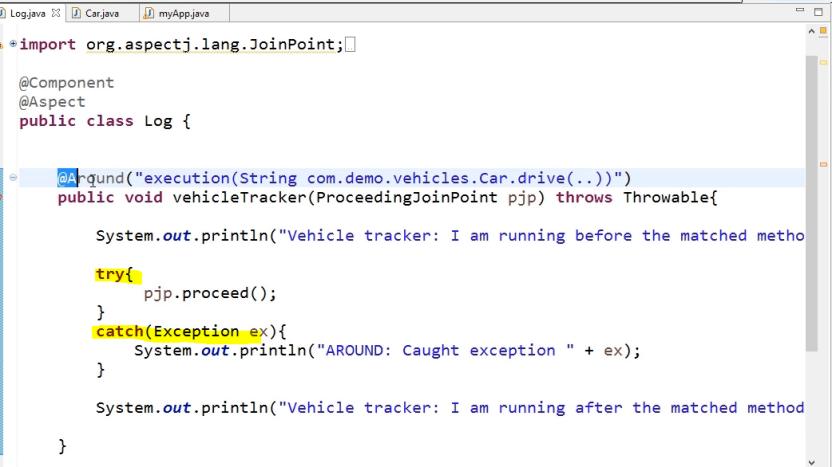


### 2.5.5 Run the Application

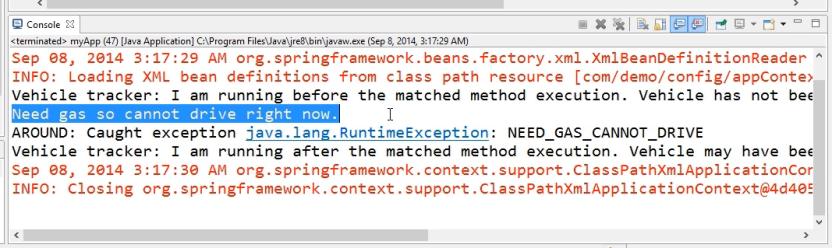


## 2.6 Working Exception with Around Advice

### 2.6.1 Create Around Advice

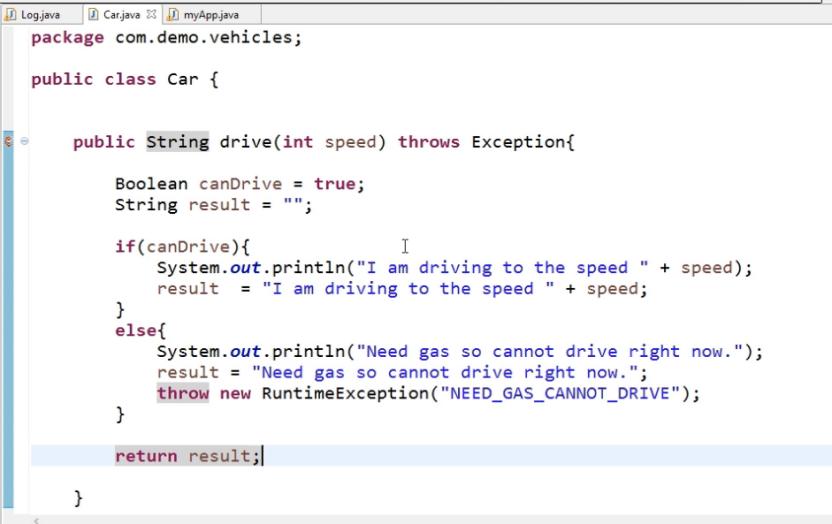


### 2.6.2 Run the Application

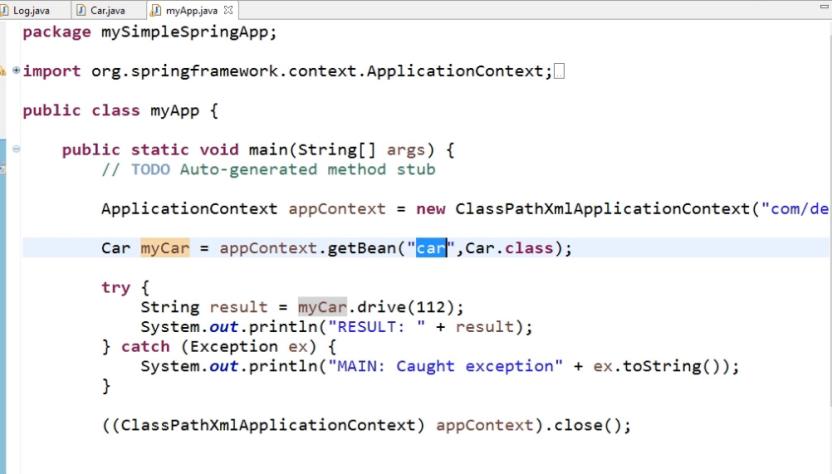


## 2.7 Working with Return Values and Parameters

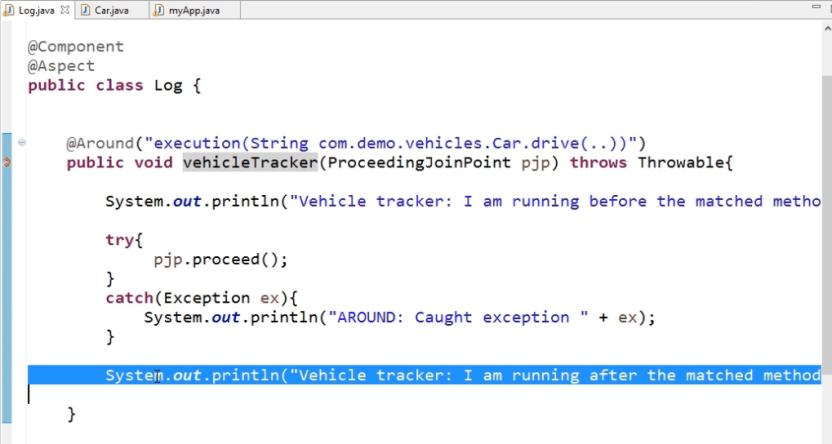
### 2.7.1 Car Class



### 2.7.2 Main App



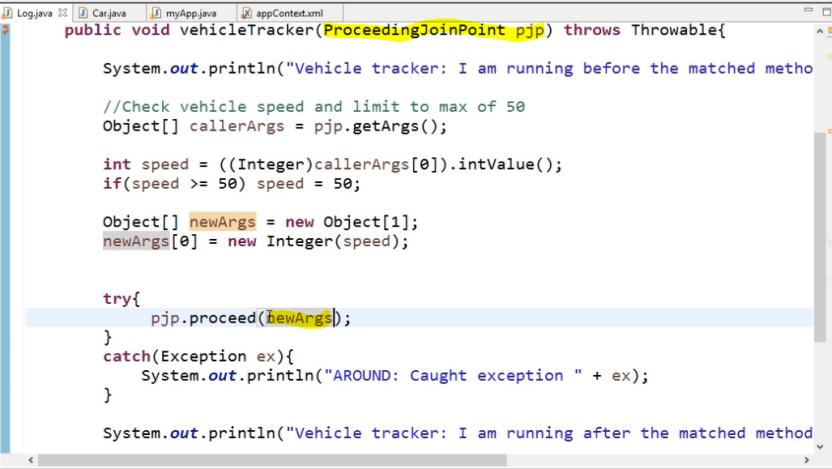
### 2.7.3 Create Around Advice



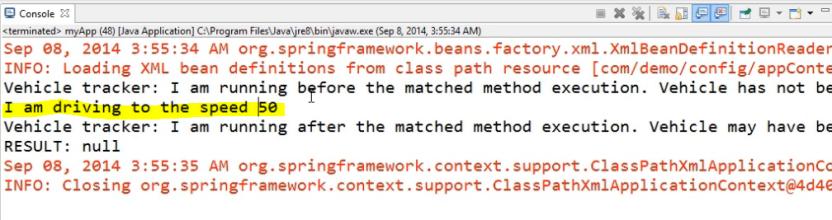
### 2.7.4 Run the Application



### 2.7.5 Override the Main Class parameter from Aspect

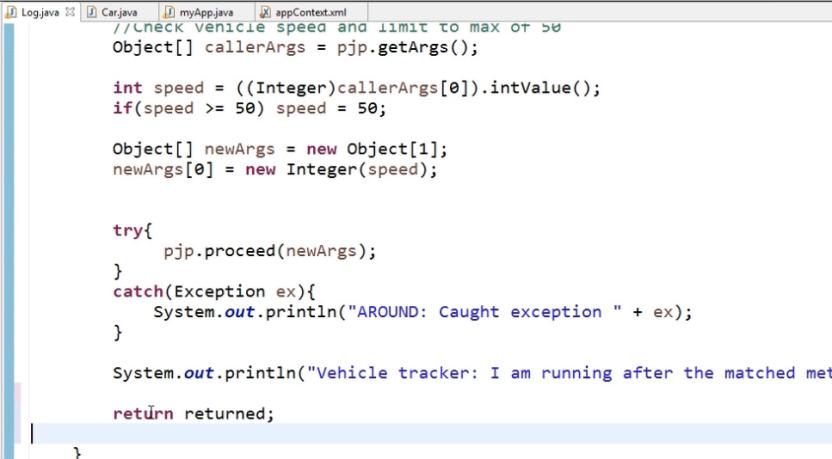


Run the application

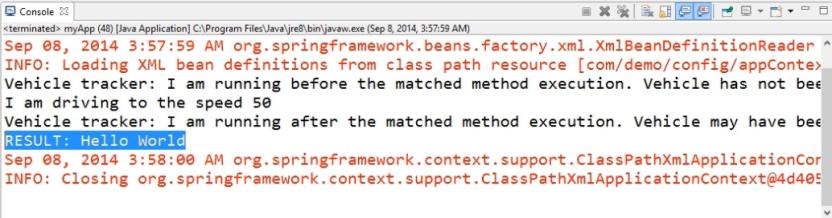


### 2.7.6 Retuning Values from Aspect class

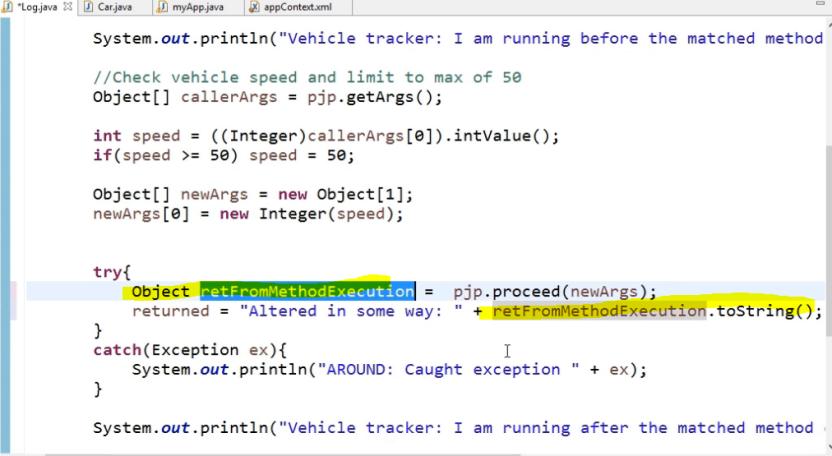




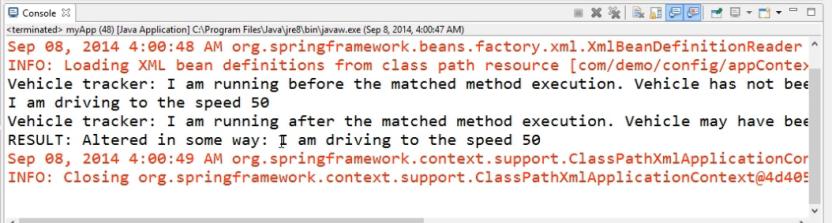
Run the Application



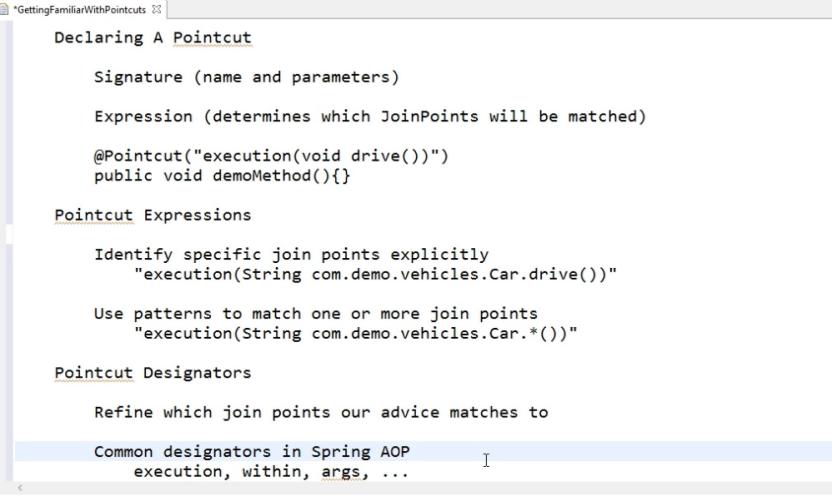
### 2.7.7 Returning values from Car Class



### 2.7.8 Run the Application

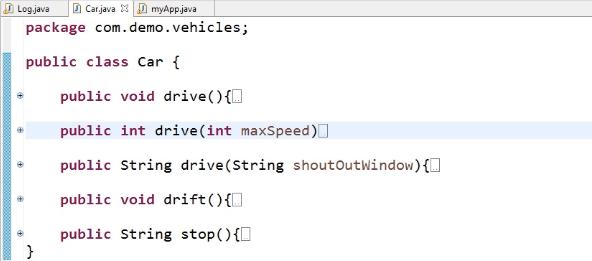


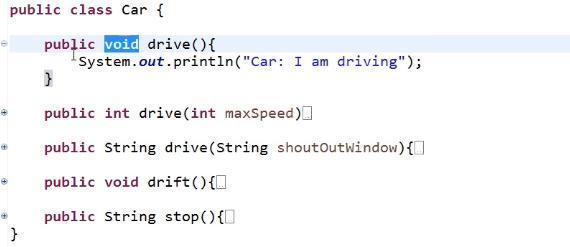
# 3. Working with Point Cut

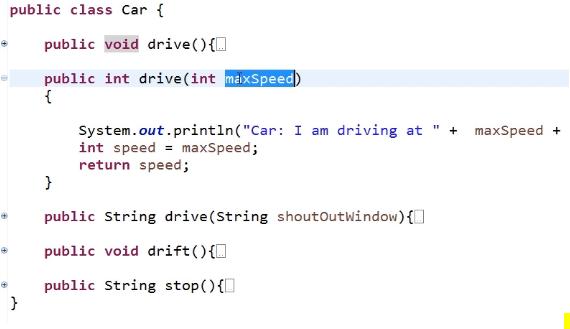


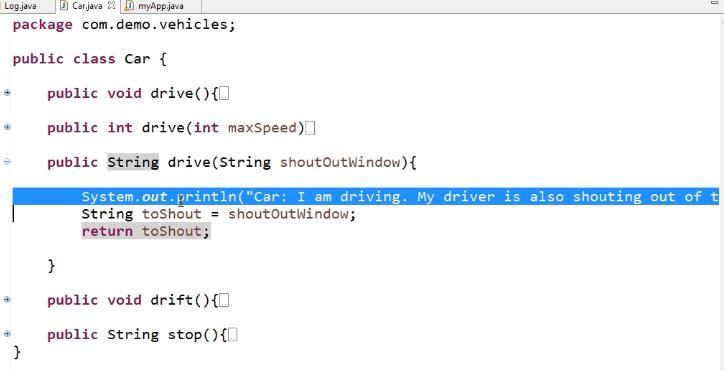
## 3.1 Combining Point Cut Expression

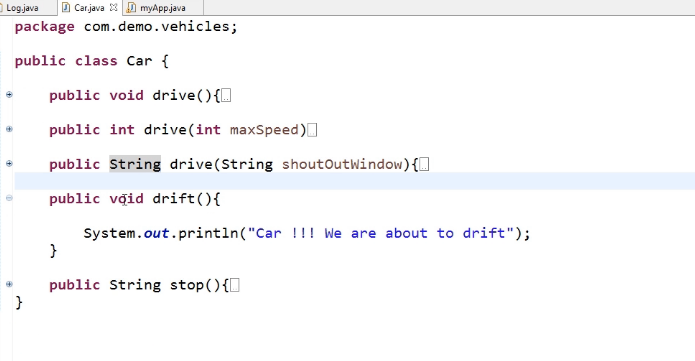
### 3.1.1 Car Class

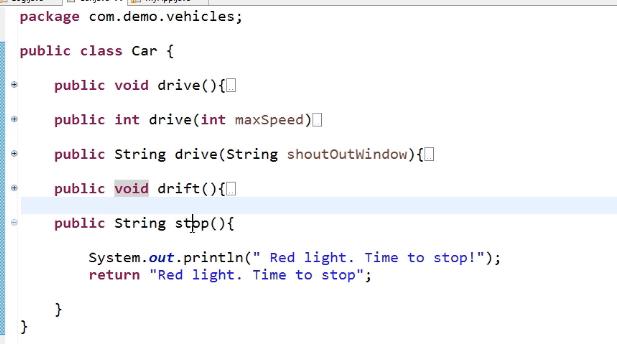






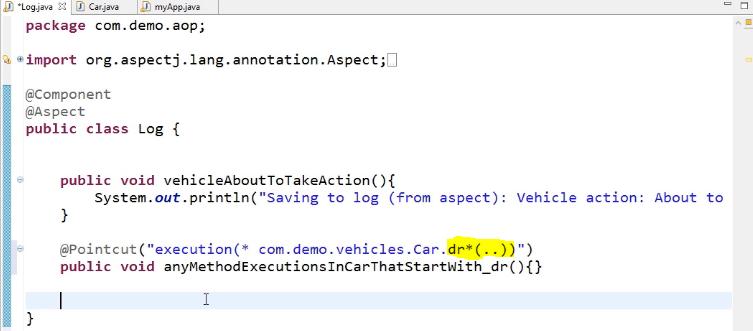




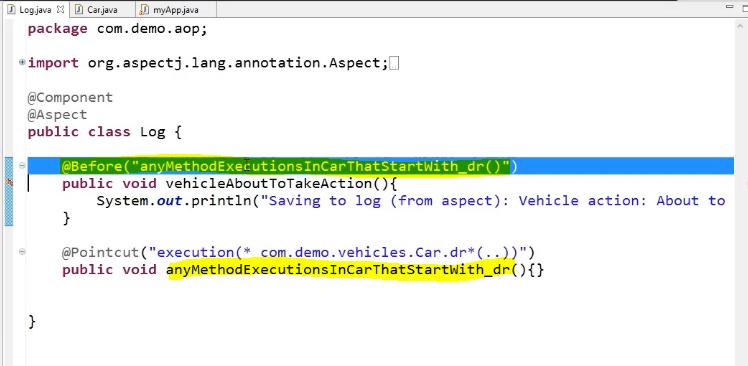


### 3.1.2 Create Point cut expression in Aspect Class

### The point cut expression will run Any method name has with dr in car class



Add the expression in before advice



### 3.1.3 Main App

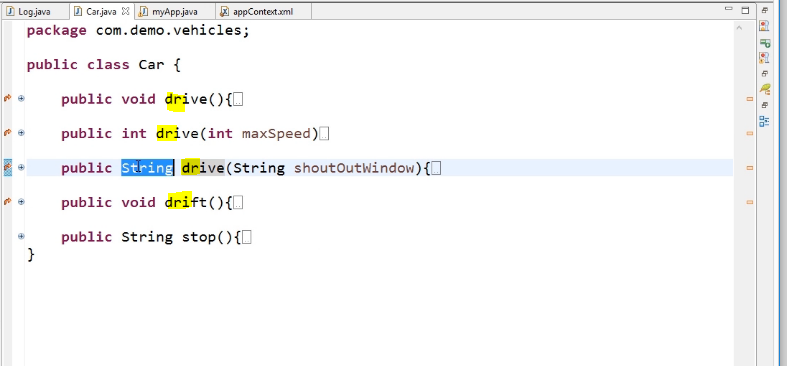


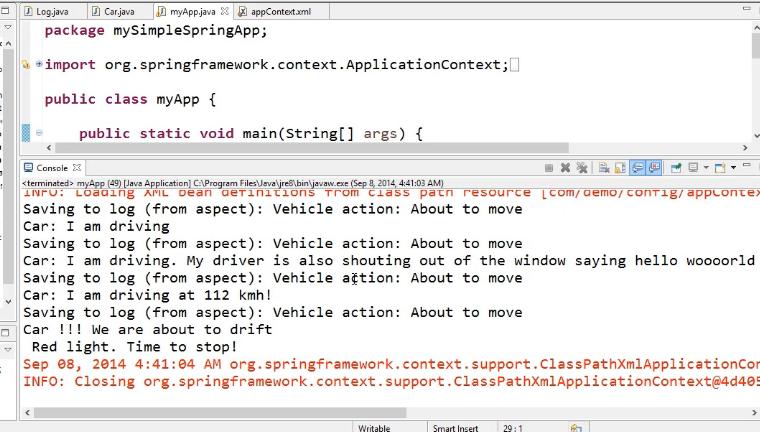
### 3.1.4 Register the Bean



### 3.1.5 Run the Application

Car class has 4 method start with dr so 4 time is execute the before advice based on point cut expression

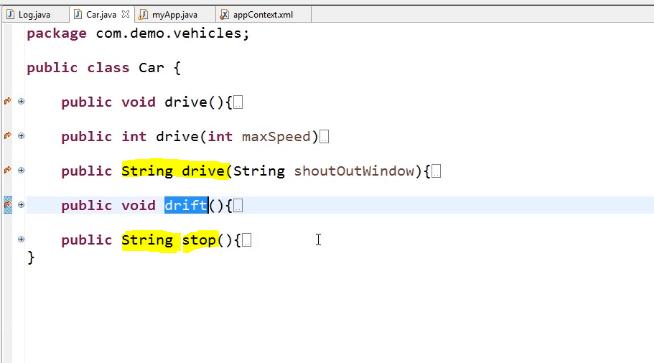


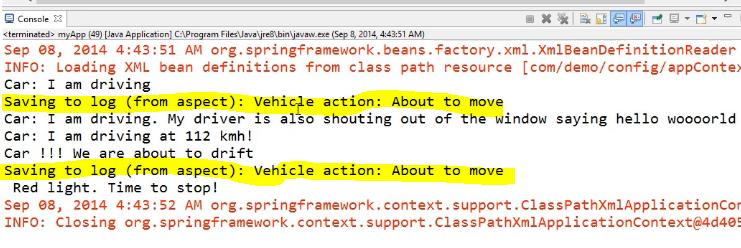


### 3.1.6 Create Point cut expression based on return type

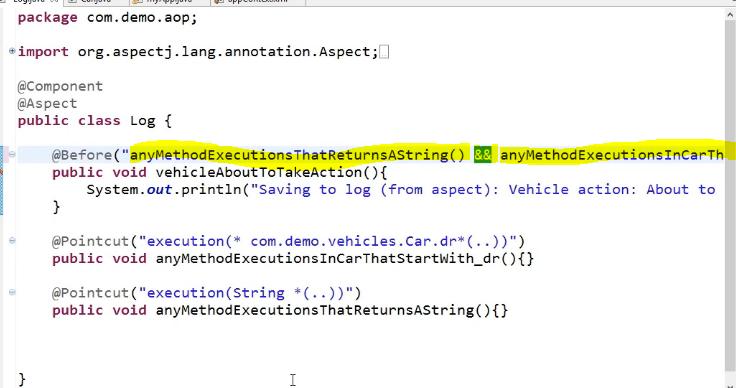


### 3.1.7 Run the application

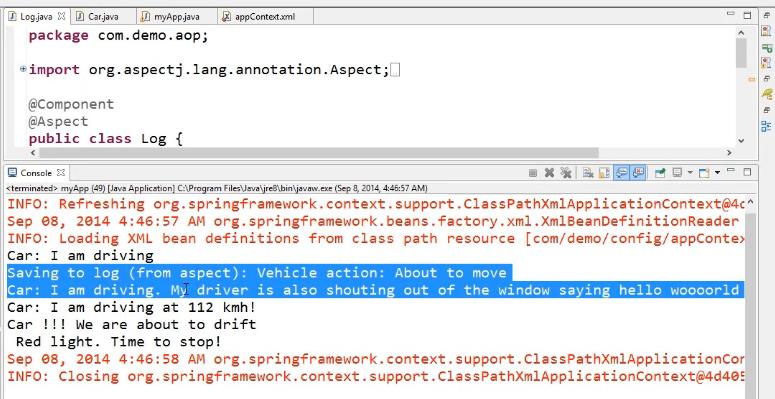




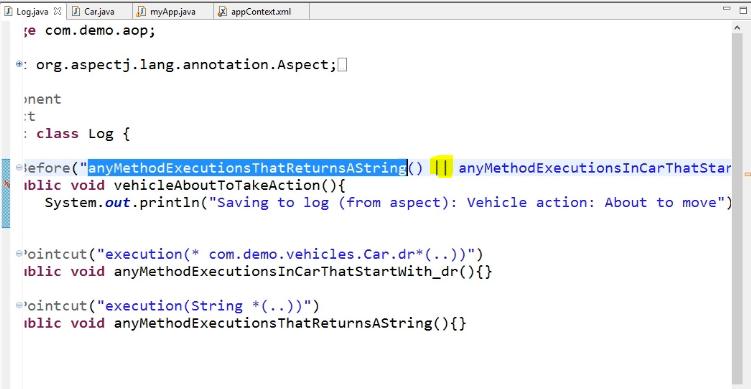
### 3.1.8 Combining Point Cut && operation



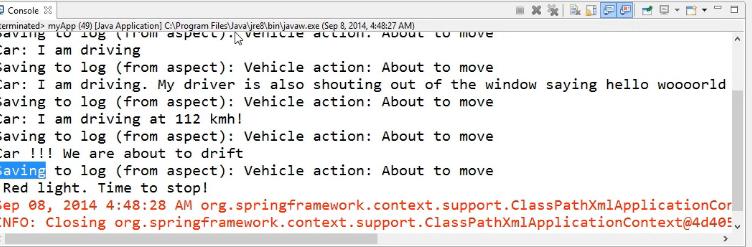
### 3.1.9 Run the Application



### 3.1.10 Combining Point Cut || operation

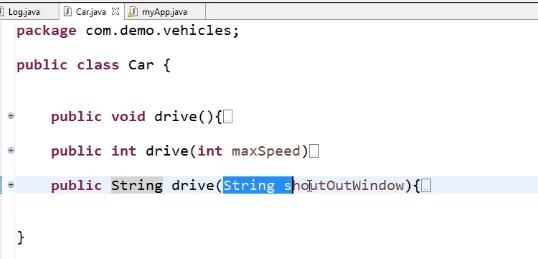


### 3.1.11 Run the Application

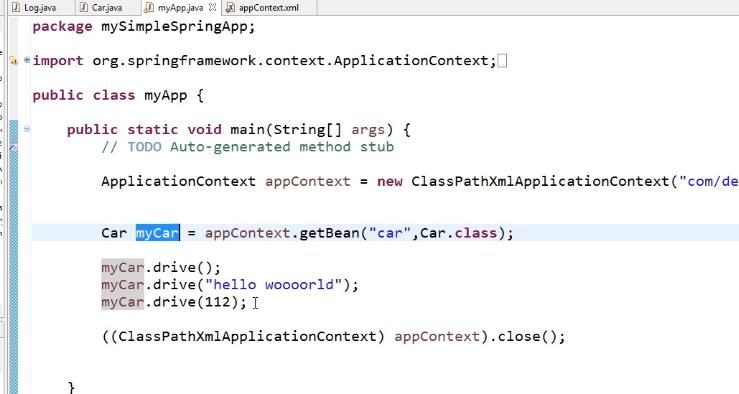


## 3.2 Matching Return Type Point Cut

### 3.2.1 Car Class



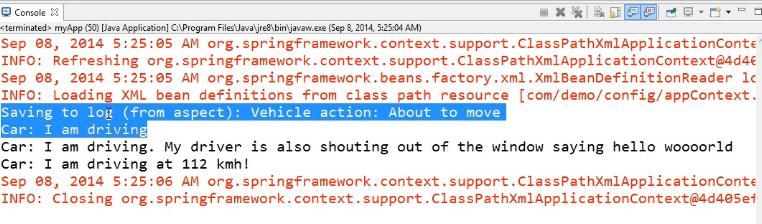
### 3.2.2 Main App



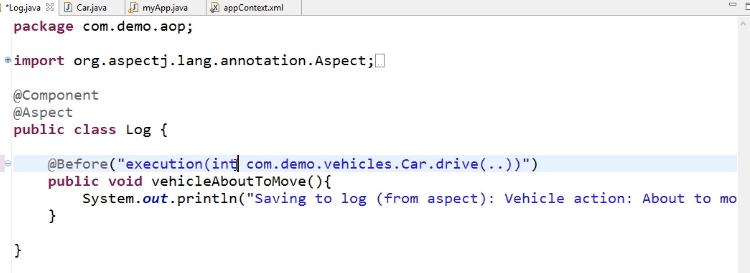
### 3.2.3 No return type Point Cut Expression



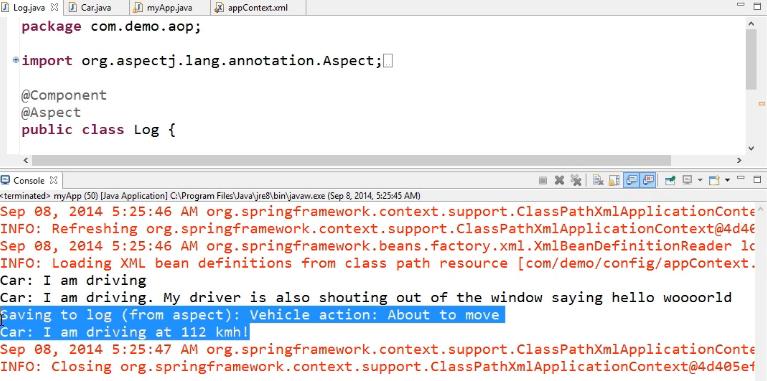
### 3.2.4 Run the Application



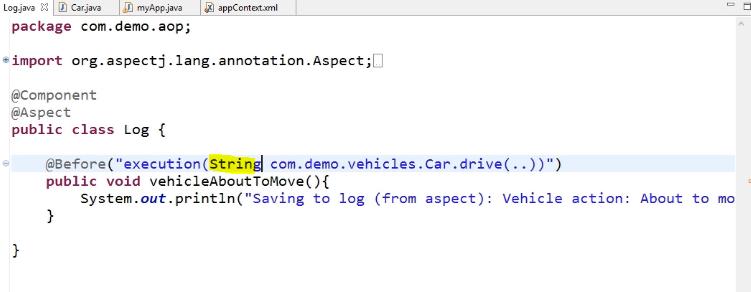
### 3.2.5 Int return type Point Cut Expression



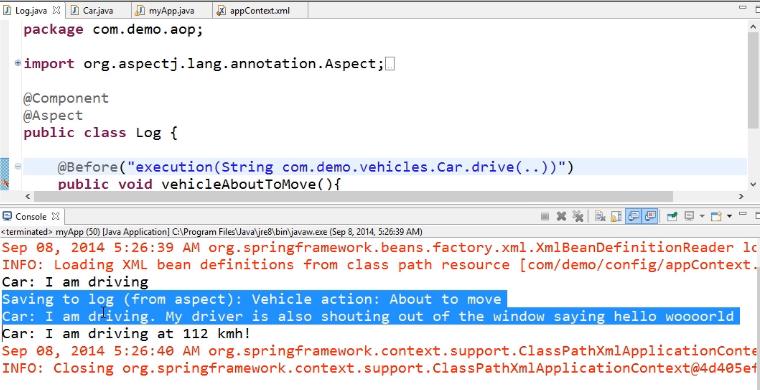
### 3.2.6 Run the Application



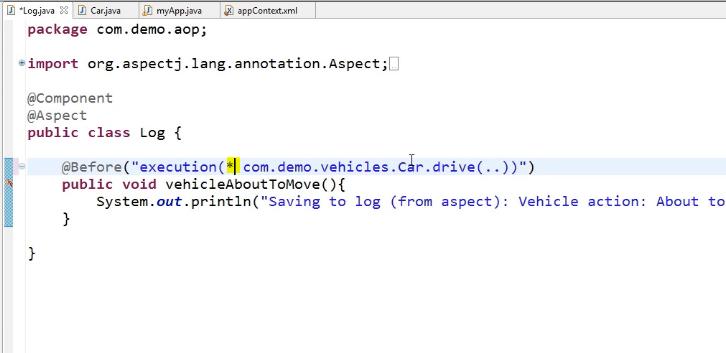
### 3.2.7 String return type Point Cut Expression



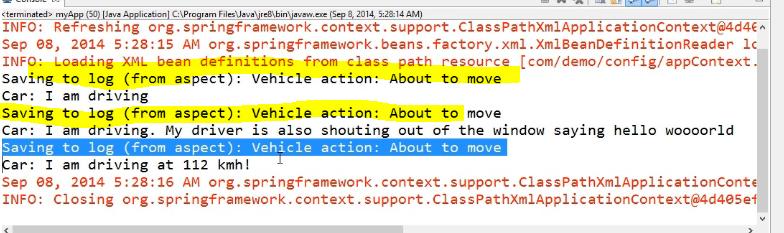
### 3.2.8 Run the application



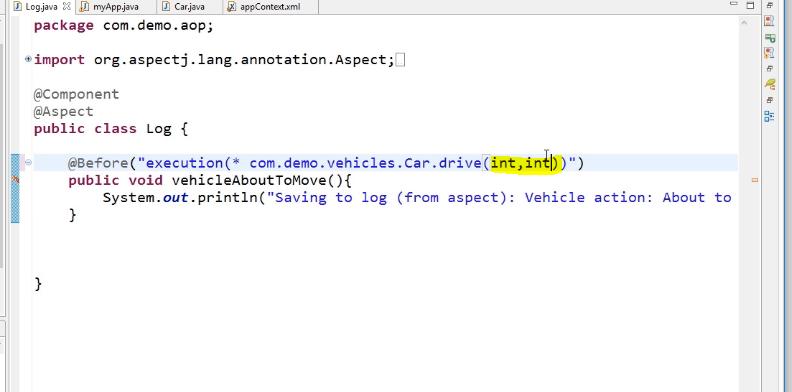
### 3.2.9 Any return type

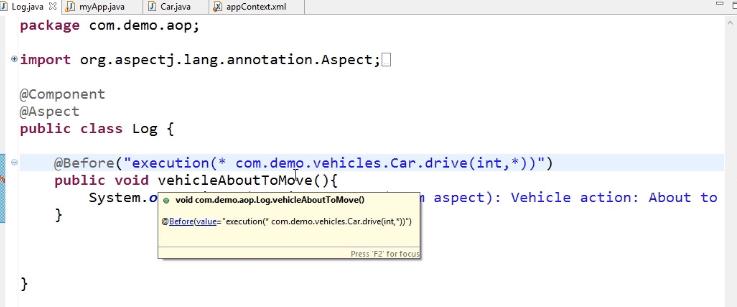


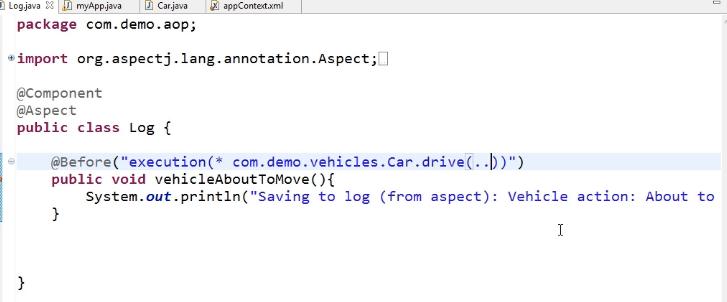
### 3.2.10 Run the Application



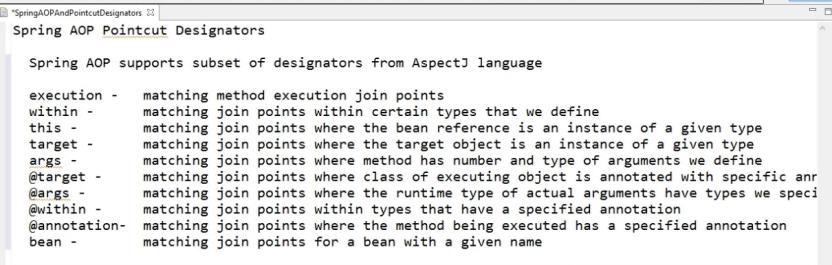
## 3.3 Matching Method Parameter Point cut Expression





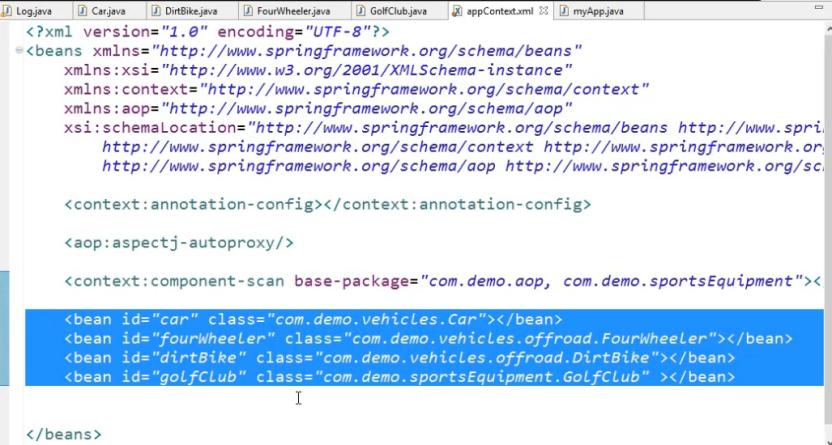


# 4. Spring AOP Point Cut Designators

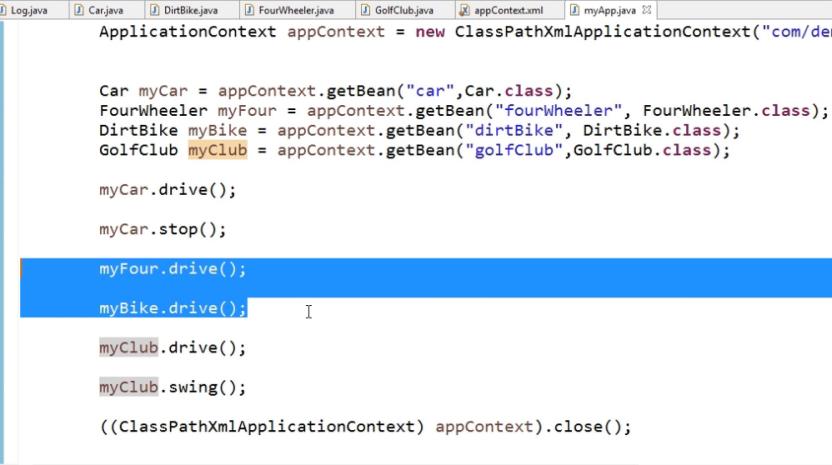


## 4.1 Matching Packages and Class Point Cut Expression

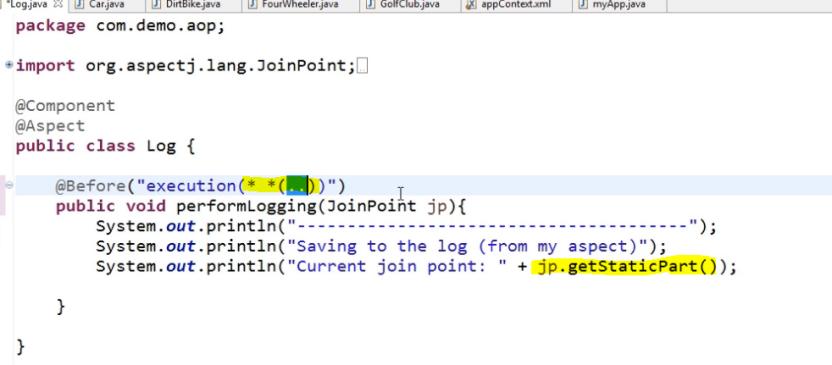
### 4.1.1 Register the bean



### 4.1.2 Main App



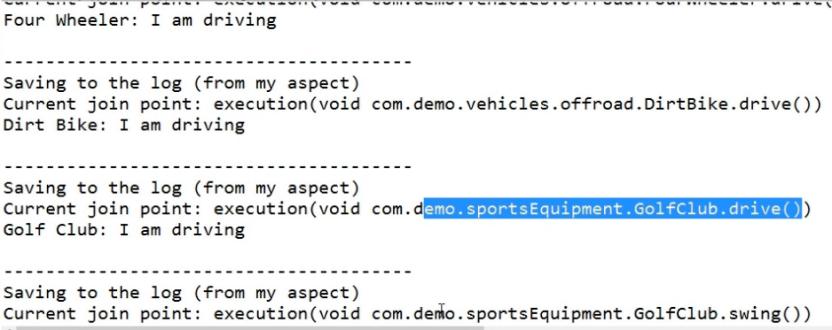
### 4.1.3 Create Point Cut Expression for Any method and Any Package and Any Return Type and Any Parameter



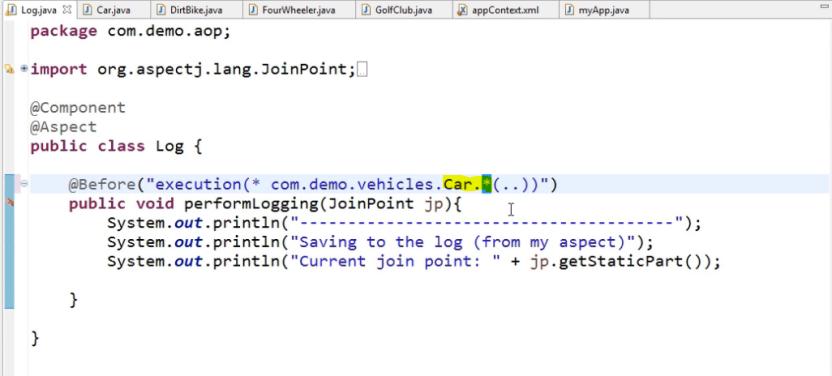
Jp.getStaticPart is return the method name

### 4.1.4 Run the application

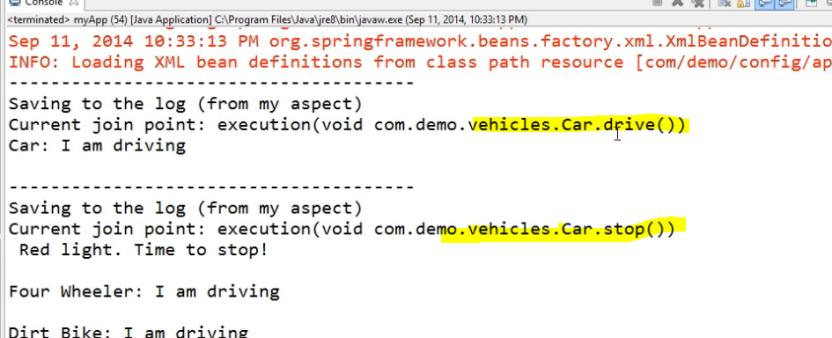




### 4.1.5 Matching the Class Point Cut Expression



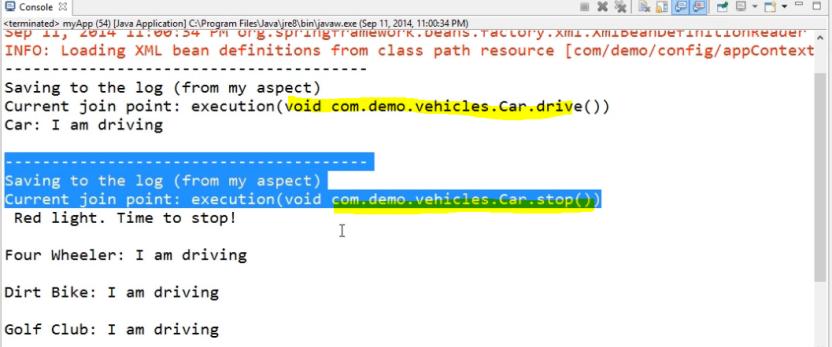
### 4.1.6 Run the Application



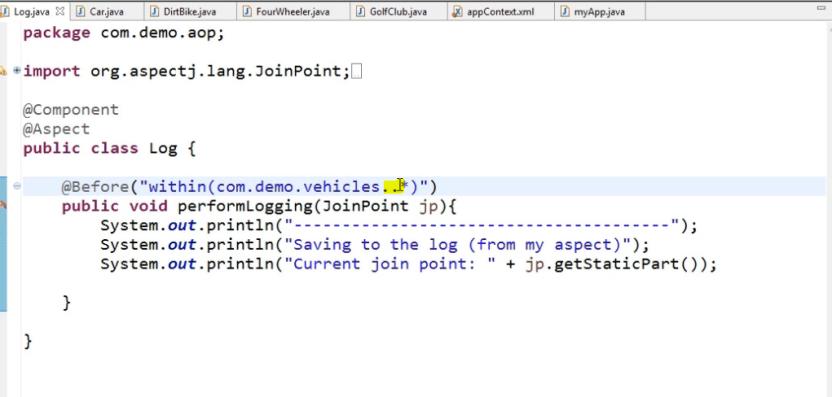
### 4.1.7 Matching the Package Point Cut Expression without Sub Packages



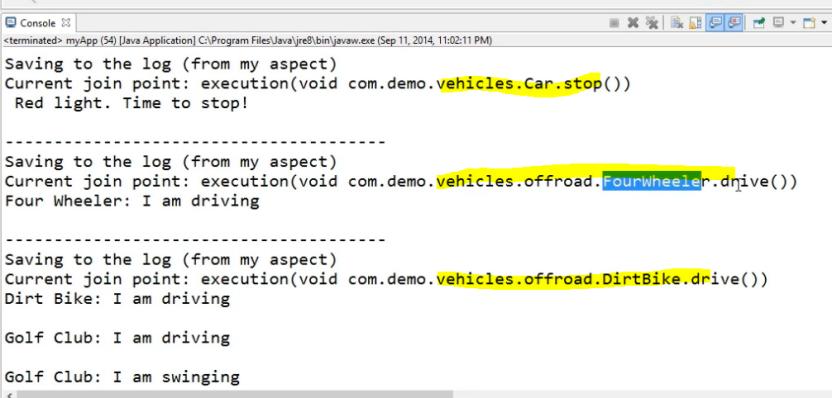
### 4.1.8 Run the Application



### 4.1.9 Matching the Package Point Cut Expression with Sub Packages

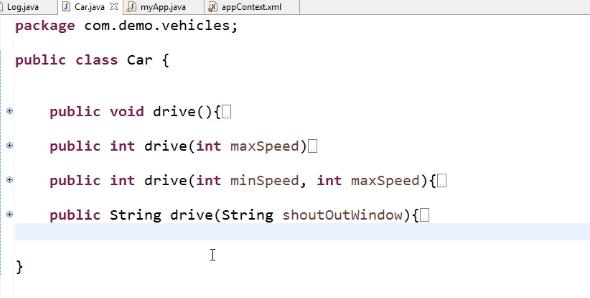


### 4.1.10 Run the Application

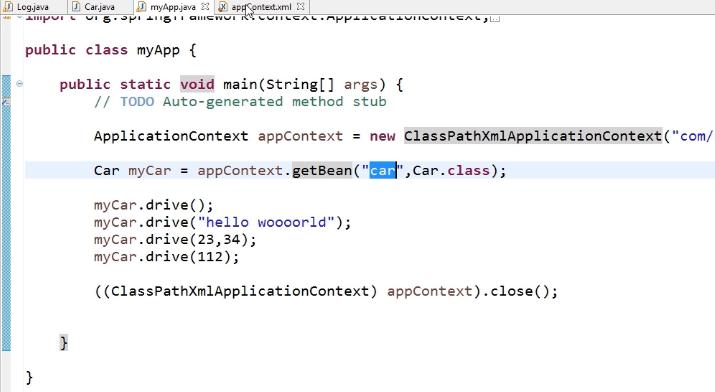


## 4.2 Using Args Point Cut Expression

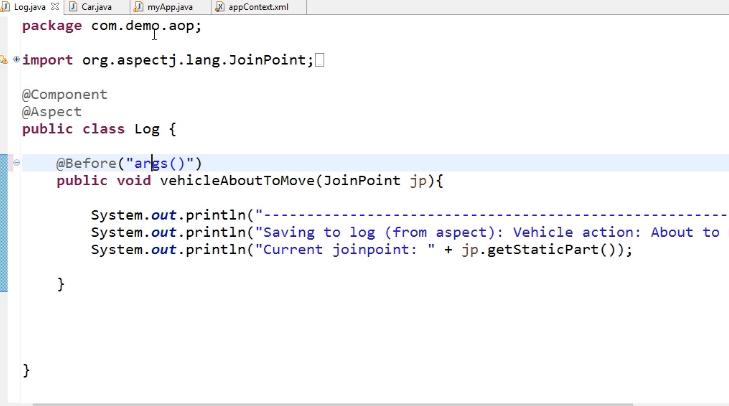
### 4.2.1 Car Class



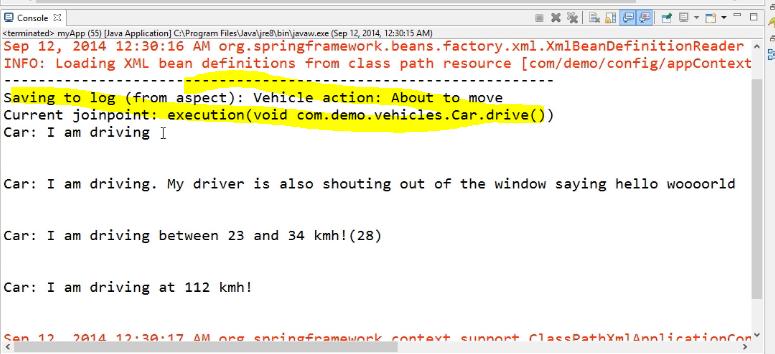
### 4.2.2 Main Class



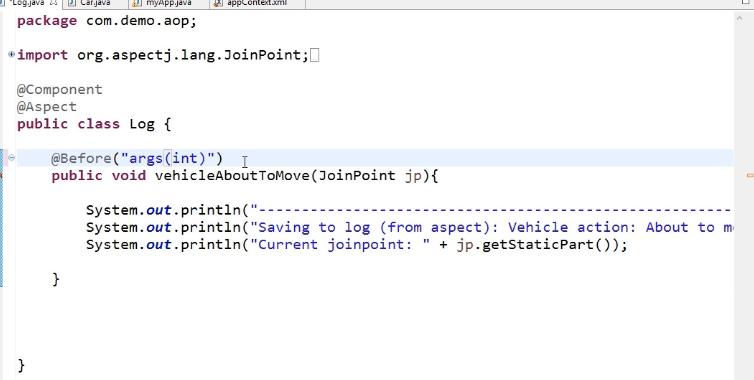
### 4.2.3 No Parameter Args Point Cut Expression



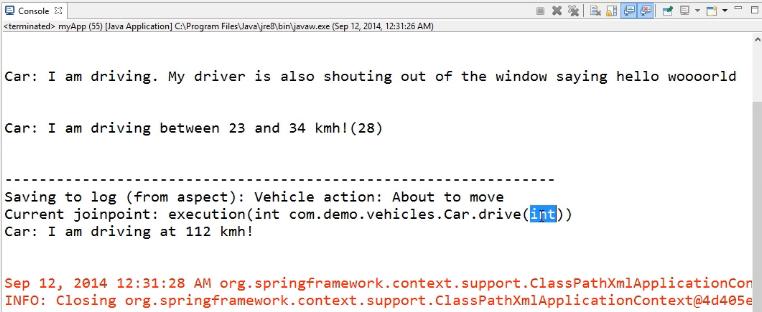
### 4.2.4 Run the Application



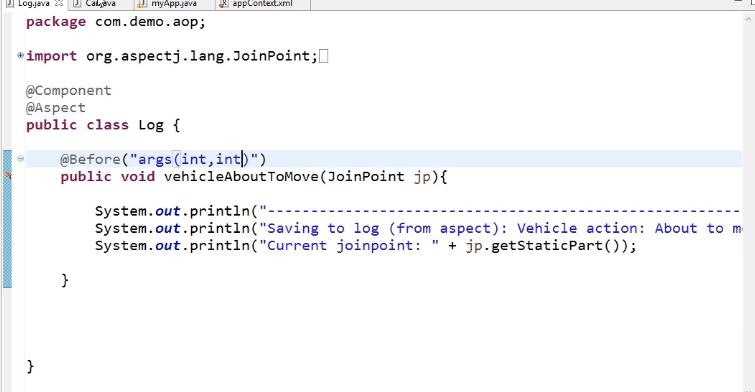
### 4.2.5 One Parameter Point Cut Expression



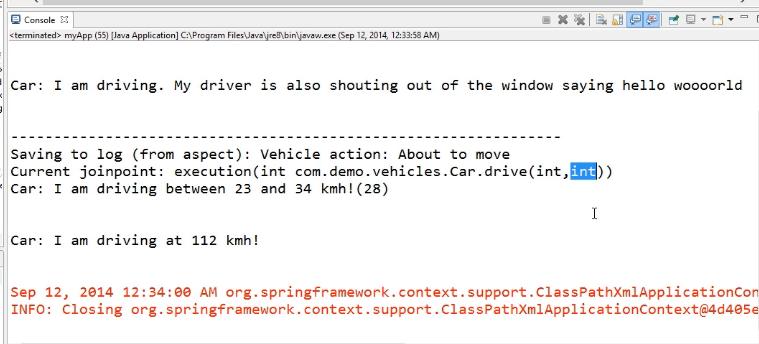
### 4.2.6 Run the Application



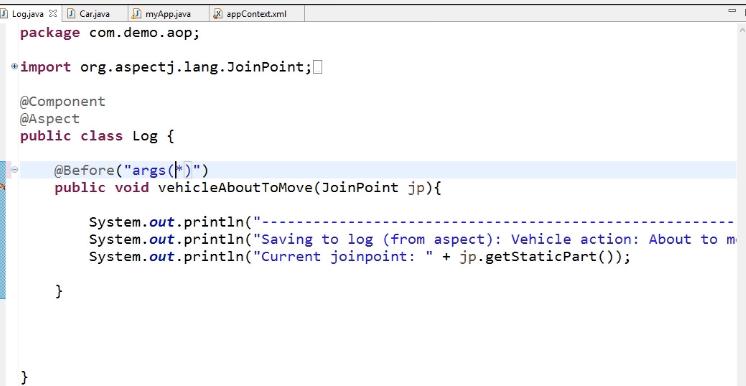
### 4.2.7 Two Parameter Point Cut Expression



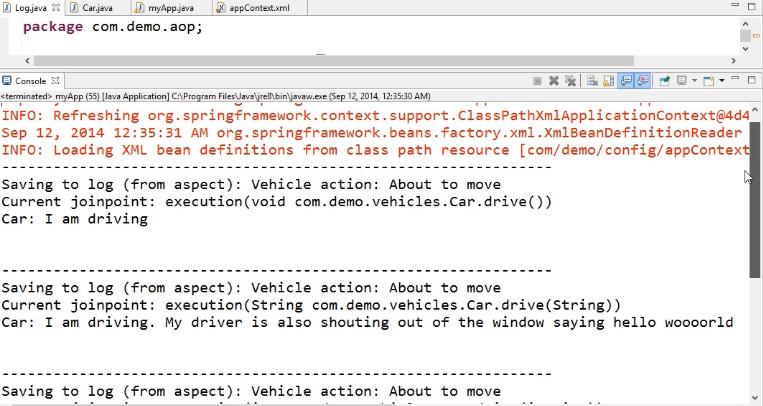
### 4.2.8 Run the Application



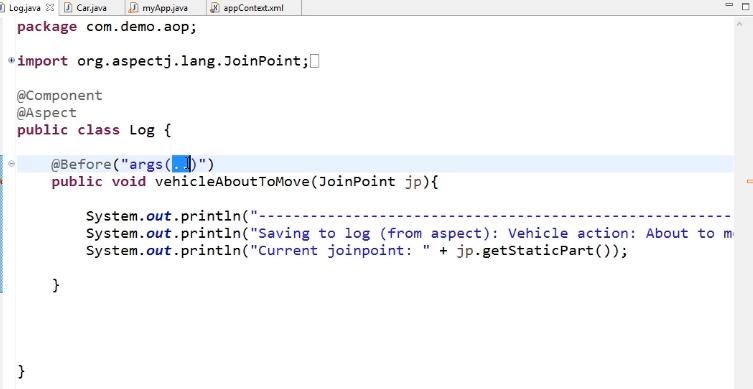
### 4.2.9 One Parameter (Any Type) Point Cut Expression



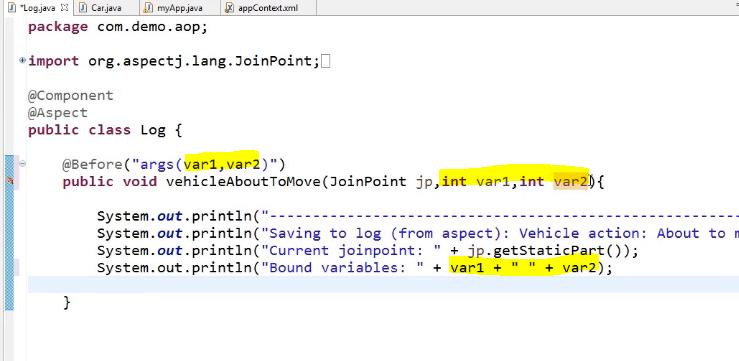
### 4.2.10 Run the Application



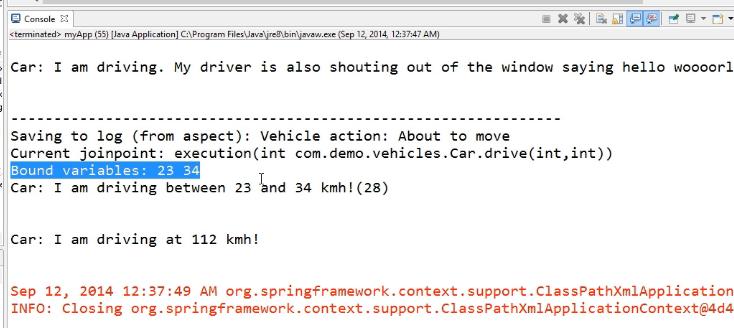
### 4.2.11 Any Number of Parameter Point Cut Expression



### 4.2.12 Read the Parameter values in Aspect Class

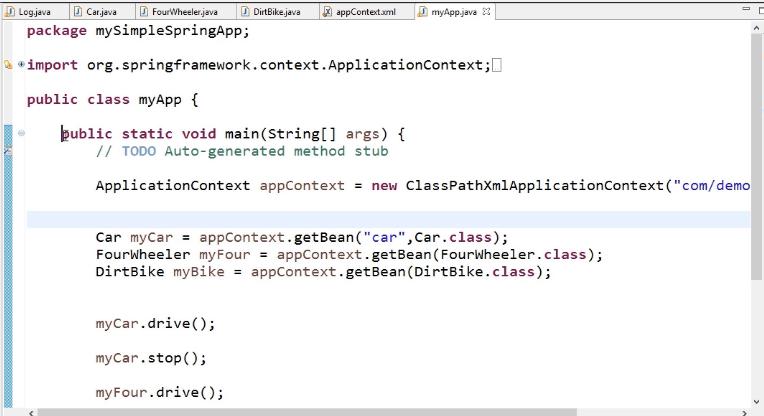


### 4.2.13 Run the Application

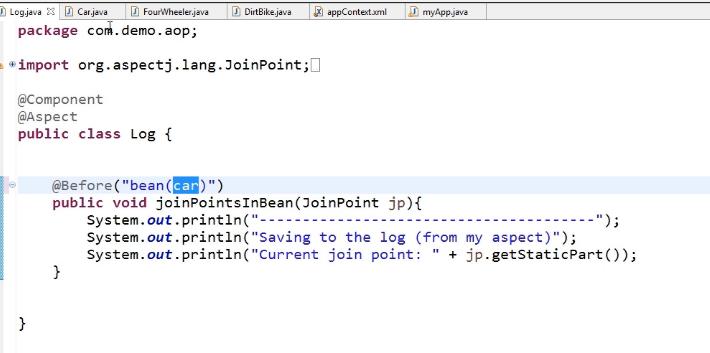


## 4.3 Using Bean Point Cut Expression

### 4.3.1 Main App



### 4.3.2 Create Point Cut Expression



### 4.3.3 Run the Application

