

Assignment No 8

Assignment based Aspect Oriented Programming

1. Write a program to demonstrate Spring AOP – before advice.

2. Write a program to demonstrate Spring AOP – after advice.

Aim: Write a program to demonstrate Spring AOP – before advice.

Write a program to demonstrate Spring AOP – after advice.

Code:

Logging.java

```
package com.aop;
public class Logging {
    public void beforeAdvice()
    {
        System.out.println("I am before advice");
    }
    public void afterAdvice()
    {
        System.out.println("I am after advice");
    }
}

package com.aop;
public class Student {
    private int id;
    private String name;
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
```

```
this.name = name;
}
public void display()
{
    System.out.println("Id is " + id);
    System.out.println(name);
}
}
```

config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:c="http://www.springframework.org/schema/c"
xmlns:cache="http://www.springframework.org/schema/cache"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:jee="http://www.springframework.org/schema/jee"
xmlns:lang="http://www.springframework.org/schema/lang"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:task="http://www.springframework.org/schema/task"
xmlns:util="http://www.springframework.org/schema/util"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
http://www.springframework.org/schema/cache
http://www.springframework.org/schema/cache/spring-cache-4.3.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd
```

```

http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee-4.3.xsd
http://www.springframework.org/schema/lang
http://www.springframework.org/schema/lang/spring-lang-4.3.xsd
http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-4.3.xsd
http://www.springframework.org/schema/util
http://www.springframework.org/schema/util/spring-util-4.3.xsd">
<bean id="stud" class="com.aop.Student">
<property name="id" value="66"></property>
<property name="name" value="Anuj Pal"></property>
</bean>
<bean id="logbean" class="com.aop.Logging"></bean>
<aop:aspectj-autoproxy></aop:aspectj-autoproxy>
<aop:config>
<aop:aspect id="log" ref="logbean">
<aop:before method="beforeAdvice" pointcut="execution(public void
display())"></aop:before>
<aop:after method="afterAdvice" pointcut="execution(public void
display())"></aop:after></aop:aspect>
</aop:config>
</beans>

```

Studentmain.java

```

package com.aop;
import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
public class StudentmainApp {
public static void main(String[] args) {

```

```
// TODO Auto-generated method stub
ApplicationContext con=new
ClassPathXmlApplicationContext("config.xml");
Student s=(Student)con.getBean("stud");
s.display();
}
}
```

Output

```
I am before advice
Id is 66
Anuj Pal
I am after advice
```

3. Write a program to demonstrate Spring AOP – around advice.

Aim: Write a program to demonstrate Spring AOP – around advice.

Code

Logging.java

```
package com.aop;
public class Logging {
    public void AroundAdvice()
    {
        System.out.println("I am around advice");
    }
}
```

config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:c="http://www.springframework.org/schema/c"
xmlns:cache="http://www.springframework.org/schema/cache"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:jee="http://www.springframework.org/schema/jee"
xmlns:lang="http://www.springframework.org/schema/lang"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:task="http://www.springframework.org/schema/task"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.3.xsd

http://www.springframework.org/schema/cache
http://www.springframework.org/schema/cache/spring-cache-4.3.xsd

http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd

http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee-4.3.xsd

http://www.springframework.org/schema/lang
http://www.springframework.org/schema/lang/spring-lang-4.3.xsd

http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-4.3.xsd">
<bean id="stud" class="com.aop.Student">
<property name="id" value="66"></property>

```
<property name="name" value="Anuj Pal"></property>

</bean>

<bean id="logbean" class="com.aop.Logging"></bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<aop:config>

<aop:aspect id="log" ref="logbean">

<aop:around method="AroundAdvice" pointcut="execution(public void
display())"></aop:around>

</aop:aspect>

</aop:config>

</beans>
```

Studentmain.java

```
package com.aop;

import org.springframework.context.ApplicationContext;

import
org.springframework.context.support.ClassPathXmlApplicationContext;

public class Studentmain {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        ApplicationContext con=new
ClassPathXmlApplicationContext("config.xml");

        Student s=(Student)con.getBean("stud");
```

```

        s.display();
    }
}

```

Output

```

I am BEFORE ADVICE
-----
I am Around ADVICE
ID : 66
Name : Anuj Pal
(-----
This is AfterReturn ADVICE
Method got successfully executed
-----)
Above data is of reuired student
-----
I am After ADVICE

```

4. Write a program to demonstrate Spring AOP – after returning advice.

Aim: Write a program to demonstrate Spring AOP – after returning advice.

Code:

Logging.java

```

package com.aop;
public class Logging {
    public void AfterReturningAdvice()
    {
        System.out.println("I am after returning advice");
    }
}

```

config.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:c="http://www.springframework.org/schema/c"
xmlns:cache="http://www.springframework.org/schema/cache"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:jee="http://www.springframework.org/schema/jee"
xmlns:lang="http://www.springframework.org/schema/lang"
xmlns:p="http://www.springframework.org/schema/p"
xmlns:task="http://www.springframework.org/schema/task"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.3.xsd

http://www.springframework.org/schema/cache
http://www.springframework.org/schema/cache/spring-cache-4.3.xsd

http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd

http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee-4.3.xsd

http://www.springframework.org/schema/lang
http://www.springframework.org/schema/lang/spring-lang-4.3.xsd

http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-4.3.xsd">
<bean id="stud" class="com.aop.Student">
```



```

<property name="id" value="66"></property>

<property name="name" value="Anuj Pal"></property>

</bean>

<bean id="logbean" class="com.aop.Logging"></bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<aop:config>

<aop:aspect id="log" ref="logbean">

<aop:after-returning method="AfterReturningAdvice"
pointcut="execution(public void display())"></aop:after-returning>

</aop:aspect>

</aop:config>

</beans>

```

Studentmain.java

```

package com.aop;
import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
public class Studentmain {
public static void main(String[] args) {
    ApplicationContext con=new
ClassPathXmlApplicationContext("config.xml");
    Student s=(Student)con.getBean("stud");
    s.display();
}
}

```

Output:

```
/tmp/aRwMNy7qlD.o  
Id is 66  
Anuj Pal  
I am after running advice
```

5. Write a program to demonstrate Spring AOP – after throwing advice.

Aim: Write a program to demonstrate Spring AOP – after throwing advice.

Code:

```
package com.aop;  
public class Logging {  
    public void AfterThrowingAdvice()  
    {  
        System.out.println("I am after returning advice");  
    }  
}
```

Student.java

```
package com.aop;  
  
public class Student {  
    private int id;  
    private String name;  
    public int getId() {  
        return id;  
    }  
    public void setId(int id) {  
        this.id = id;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

```
public void display()
{
    System.out.println("Id is "+ id);
    System.out.println("Name is "+ name);
}
}
```

Config.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:aop="http://www.springframework.org/schema/aop"
        xmlns:c="http://www.springframework.org/schema/c"
        xmlns:cache="http://www.springframework.org/schema/cache"
        xmlns:context="http://www.springframework.org/schema/context"
        xmlns:jee="http://www.springframework.org/schema/jee"
        xmlns:lang="http://www.springframework.org/schema/lang"
        xmlns:p="http://www.springframework.org/schema/p"
        xmlns:task="http://www.springframework.org/schema/task"

        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
        http://www.springframework.org/schema/cache
http://www.springframework.org/schema/cache/spring-cache-4.3.xsd
```

<http://www.springframework.org/schema/context>
<http://www.springframework.org/schema/context/spring-context-4.3.xsd>

<http://www.springframework.org/schema/jee>
<http://www.springframework.org/schema/jee/spring-jee-4.3.xsd>

<http://www.springframework.org/schema/lang>
<http://www.springframework.org/schema/lang/spring-lang-4.3.xsd>

<http://www.springframework.org/schema/task>
[http://www.springframework.org/schema/task/spring-task-4.3.xsd"](http://www.springframework.org/schema/task/spring-task-4.3.xsd)>

```
<bean id="stud" class="com.aop.Student">
<property name="id" value="66"></property>
<property name="name" value="Anuj Pal "></property>
</bean>

<bean id="logbean" class="com.aop.Logging"></bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<aop:config>

<aop:aspect id="log" ref="logbean">

<aop:after-throwing method="AfterThrowingAdvice"
pointcut="execution(public void display())"></aop:after-throwing>

</aop:aspect>

</aop:config>

</beans>
```

Studentmain.java

```
package com.aop;
```

```

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
public class Studentmain {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext con=new
ClassPathXmlApplicationContext("config.xml");
        Student s=(Student)con.getBean("stud");
        s.display();
    }
}

```

Output:

```

This is After Throwing Advice
Exception in thread "main" java.lang.IllegalArgumentException
at com.aop.Student.display(Student.java:21)
at com.aop.Student$$FastClassBySpringCGLIB$$70d78142.invoke(<generated>)
at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:218)
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpoint(CglibAopProxy.java:783)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:163)
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:753)
at org.springframework.aop.aspectj.AspectJAfterThrowingAdvice.invoke(AspectJAfterThrowingAdvice.java:64)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:186)
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:753)
at org.springframework.aop.interceptor.ExposeInvocationInterceptor.invoke(ExposeInvocationInterceptor.java:97)
at org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMethodInvocation.java:186)
at org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed(CglibAopProxy.java:753)

```

6. Write a program to demonstrate Spring AOP – pointcuts

Aim: Write a program to demonstrate Spring AOP – pointcuts

Code

Student.java

```

package com.aop;
public class Student {
    private int id;
    private String name;
    public int getId() {

```

```

        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void display()
    {
        System.out.println("Id is "+ id);
        System.out.println("Name is "+ name);
    }
}

```

Logging.java

```

package com.aop;
public class Logging {
    public void Pointcuts()
    {
        System.out.println("Pointcuts advice");
    }
}

```

Config.xml

```

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:aop="http://www.springframework.org/schema/aop"
        xmlns:c="http://www.springframework.org/schema/c"

```

xmlns:cache="http://www.springframework.org/schema/cache"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:jee="http://www.springframework.org/schema/jee"

xmlns:lang="http://www.springframework.org/schema/lang"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:task="http://www.springframework.org/schema/task"

xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-4.3.xsd

http://www.springframework.org/schema/cache
http://www.springframework.org/schema/cache/spring-cache-4.3.xsd

http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd

http://www.springframework.org/schema/jee
http://www.springframework.org/schema/jee/spring-jee-4.3.xsd

http://www.springframework.org/schema/lang
http://www.springframework.org/schema/lang/spring-lang-4.3.xsd

http://www.springframework.org/schema/task
http://www.springframework.org/schema/task/spring-task-4.3.xsd">

<bean id="stud" class="com.aop.Student">

<property name="id" value="66"></property>

<property name="name" value="Anuj Pal"></property>

</bean>

```

<bean id="logbean" class="com.aop.Logging"></bean>

<aop:aspectj-autoproxy></aop:aspectj-autoproxy>

<aop:config>

<aop:aspect id="log" ref="logbean">

<aop:pointcut id="pointCut" expression="execution(public void
display())"></aop:pointcut>

</aop:aspect>

</aop:config>

</beans>

```

Studentmain.java

```

package com.aop;
import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationContext;
public class Studentmain {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext con=new
ClassPathXmlApplicationContext("config.xml");
        Student s=(Student)con.getBean("stud");
        s.display();
    }
}

```

Output:

```

/tmp/aRWMNy7qlD.o
Id is 66
Anuj Pal
|

```


