

## Susmit Naik 35

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

### BeforeAopSusmit.java

```
package bvimit.edu;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.aspectj.lang.annotation.Pointcut;
@Aspect
public class BeforeAopSusmit{
    @Pointcut("execution(int BeforeOperationSusmit.*(..))")
    public void p(){}
    @Before("p()")
    public void myAdvice(JoinPoint jp){
        System.out.println("before advice");
    }
}
```

### BeforeOperationSusmit.java

```
package bvimit.edu;
public class BeforeOperationSusmit{
    public void msg(){System.out.println("method 1");}
    public int m(){System.out.println("method 2 with return");return 2;}
    public int k(){System.out.println("method 3 with return");return 3;}
}
```

### aopctx1.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"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="opBean" class="bvimit.edu.BeforeOperationSusmit"/>
    <bean id="trackMyBean" class="bvimit.edu.BeforeAopSusmit"/>
    <bean
        class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCrea
        tor"/>
</beans>
```

### **BeforeTestSusmit.java**

```
package bvimit.edu;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class BeforeTestSusmit{
public static void main(String[] args){
ApplicationContext context=new ClassPathXmlApplicationContext("aopctx1.xml");
BeforeOperationSusmit e=(BeforeOperationSusmit)context.getBean("opBean");
System.out.println("calling m1.....");
e.msg();
System.out.println("calling m2.....");
e.m();
System.out.println("calling m3.....");
e.k();
}
}
```

### **Output**

```
calling m1.....  
method 1  
calling m2.....  
before advice  
method 2 with return  
calling m3.....  
before advice  
method 3 with return
```

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

### **AfterAopDataSusmit.java**

```
package bvimit.edu;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Pointcut;
@Aspect
public class AfterAopDataSusmit{
@Pointcut("execution(int AfterOperationSusmit.*(..))")
public void p(){}
@After("p()")
public void myadvice(JoinPoint jp){
System.out.println("after advice");
}
}
```

### **AfterOperationSusmit.java**

```
package bvimit.edu;
public class AfterOperationSusmit{
    public void msg(){System.out.println("method 1");}
    public int m(){System.out.println("method 2 with return");return 2;}
    public int k(){System.out.println("method 3 with return");return 3;}
}
```

### **aopctx.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"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="opBean" class="bvimit.edu.AfterOperationSusmit"/>
    <bean id="trackMyBean" class="bvimit.edu.AfterAopDataSusmit"/>
    <bean
        class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator"/>
</beans>
```

### **AfterTestSusmit.java**

```
package bvimit.edu;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class AfterTestSusmit{
    public static void main(String[] args){
        ApplicationContext context=new ClassPathXmlApplicationContext("aopctx.xml");
        AfterOperationSusmit e=(AfterOperationSusmit)context.getBean("opBean");
        System.out.println("calling m1.....");
        e.msg();
        System.out.println("calling m2.....");
        e.m();
        System.out.println("calling m3.....");
        e.k();
    }
}
```

### **Output**

```
calling m1.....  
method 1  
calling m2.....  
method 2 with return  
after advice  
calling m3.....  
method 3 with return  
after advice
```

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

#### **BankaopdataSusmit.java**

```
package bvimit.edu;  
import org.aspectj.lang.ProceedingJoinPoint;  
import org.aspectj.lang.annotation.Around;  
import org.aspectj.lang.annotation.Aspect;  
import org.aspectj.lang.annotation.Pointcut;  
@Aspect  
public class BankaopdataSusmit{  
    @Pointcut("execution(* BankSusmit.*(..))")  
    public void a(){}  
    @Around("a()")  
    public Object myadvice(ProceedingJoinPoint p) throws Throwable{  
        System.out.println("Around concern Before calling actual method");  
        Object obj=p.proceed();  
        System.out.println("Around Concern After calling actual method");  
        return obj;  
    }  
}
```

#### **BankSusmit.java**

```
package bvimit.edu;  
public class BankSusmit{  
    public void welcome(){System.out.println("welcome to bank");}  
    public int icici(){System.out.println("icici bank interest rate");return 7;}  
    public int pnb(){System.out.println("pnb bank interest rate");return 6;}  
}
```

### **Bankaopdata.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"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd">
    <bean id="opBean" class="bvimit.edu.BankSusmit"/>
    <bean id="trackMyBean" class="bvimit.edu.BankaopdataSusmit"/>
    <bean
        class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator"/>
</beans>
```

### **BanktestSusmit.java**

```
package bvimit.edu;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class BanktestSusmit{
    private static ApplicationContext context;
    public static void main(String[] args){
        context=new ClassPathXmlApplicationContext("Bankaopdata.xml");
        BankSusmit e=(BankSusmit)context.getBean("opBean");
        System.out.println("Calling welcome method...");
        e.welcome();
        System.out.println("Calling icici method...");
        e.icici();
        System.out.println("Calling pnb method...");
        e.pnb();
    }
}
```

### **Output**

---

```
Around concern Before calling actual method
welcome to bank
Around Concern After calling actual method
Around concern Before calling actual method
icici bank interest rate
Around Concern After calling actual method
Around concern Before calling actual method
pnb bank interest rate
Around Concern After calling actual method
```

---

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

### **BankaopdataSusmit.java**

```
package bvimit.edu;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.AfterReturning;
import org.aspectj.lang.annotation.Aspect;
@Aspect
public class BankaopdataSusmit{
@AfterReturning(pointcut="execution(* BankSusmit.*(..))",returning="result")
public void myadvice(JoinPoint jp, Object result){
System.out.println("AfterReturning concern");
System.out.println("Result in advice"+result);
}
}
```

### **BankSusmit.java**

```
package bvimit.edu;
public class BankSusmit{
public void welcome(){System.out.println("welcome to bank");}
public int icici(){System.out.println("icici bank interest rate");return 7;}
public int pnb(){System.out.println("pnb bank interest rate");return 6;}
}
```

### **Bankaopdata.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"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="bvimit.edu.BankSusmit"/>
<bean id="trackMyBean" class="bvimit.edu.BankaopdataSusmit"/>
<bean
class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator"/>
</beans>
```

### **BanktestSusmit.java**

```
package bvimit.edu;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class BanktestSusmit{
private static ApplicationContext context;
public static void main(String[] args){
context=new ClassPathXmlApplicationContext("Bankaopdata.xml");
BankSusmit e=(BankSusmit)context.getBean("opBean");
e.welcome();
e.icici();
e.pnb();
}
}
```

```
welcome to bank
AfterReturning concern
Result in advice null
icici bank interest rate
AfterReturning concern
Result in advice 7
pnb bank interest rate
AfterReturning concern
Result in advice 6
```

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

### **Operationaop\_atSusmit.java**

```
package bvimit.edu;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.AfterThrowing;
import org.aspectj.lang.annotation.Aspect;
@Aspect
public class Operationaop_atSusmit{
@AfterThrowing(pointcut="execution(* Operation_atSusmit.*(..))",throwing="error")
public void myadvice(JoinPoint jp,Throwable error){
System.out.println("AfterThrowing concern");
System.out.println("Exception is: "+error);
System.out.println("end of after throwing advice....");
}
}
```

### **Operation\_atSusmit.java**

```
package bvimit.edu;
public class Operation_atSusmit{
public void validate(int att) throws Exception{
if(att<75){
throw new ArithmeticException("Not eligible for exam");
} else{
System.out.println("Eligible for exam");
}
}
}
```

### **validctx.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"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="bvimit.edu.Operation_atSusmit"/>
<bean id="trackMyBean" class="bvimit.edu.Operationaop_atSusmit"/>
<bean
class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator"/>
</beans>
```

### **OperationTest\_atSusmit.java**

```
package bvimit.edu;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class OperationTest_atSusmit{
private static ApplicationContext context;
public static void main(String[] args){
ApplicationContext context=new ClassPathXmlApplicationContext("validctx.xml");
Operation_atSusmit op=(Operation_atSusmit)context.getBean("opBean");
System.out.println("calling validate....");
try{
op.validate(85);
} catch(Exception e){System.out.println(e);}
System.out.println("calling validate again....");
try{
op.validate(25);
} catch(Exception e){System.out.println(e);}
}
```

```
}
```

## Output

```
calling validate....
Eligible for exam
calling validate again....
AfterThrowing concern
Exception is: java.lang.ArithmeticException: Not eligible for exam
end of after throwing advice....
java.lang.ArithmeticException: Not eligible for exam
```

**Write a program to demonstrate Spring AOP –pointcuts.**

### Operation\_pcSusmit.java

```
package bvimit.edu;
public class Operation_pcSusmit{
    public void msg(){System.out.println("method 1");}
    public int m(){System.out.println("method 2 with return");return 2;}
    public int k(){System.out.println("method 3 with return");return 3;}
}
```

### Aopdata\_pcSusmit.java

```
package bvimit.edu;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.Pointcut;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
@Aspect
public class Aopdata_pcSusmit{
    @Pointcut("execution(int Operation_pcSusmit.*(..))")
    public void p(){}
    @After("p()")
    public void myadvice(JoinPoint jp){
        System.out.println("After advice");
    }
    @Pointcut("execution(* Operation_pcSusmit.*(..))")
    public void i(){}
    @Before("i()")
    public void myadvice1(JoinPoint jp){
        System.out.println("Before advice");
    }
}
```

```
}
```

### **Test\_pcSusmit.java**

```
package bvimit.edu;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Test_pcSusmit{
public static void main(String[] args){
ApplicationContext context=new ClassPathXmlApplicationContext("aopctx_pc.xml");
Operation_pcSusmit e=(Operation_pcSusmit)context.getBean("opBean");
System.out.println("calling m1...");
e.msg();
System.out.println("calling m2...");
e.m();
System.out.println("calling m3...");
e.k();
}
}
```

### **aopctx\_pc.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"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
<bean id="opBean" class="bvimit.edu.Operation_pcSusmit"/>
<bean id="trackMyBean" class="bvimit.edu.Aopdata_pcSusmit"/>
<bean
class="org.springframework.aop.aspectj.annotation.AnnotationAwareAspectJAutoProxyCreator"/>
</beans>
```

### **Output**

```
calling m1...
method 1
calling m2...
method 2 with return
calling m3...
method 3 with return
```