

Program no. 1 Write a program to demonstrate Spring AOP – before advice..

Logging.java

```
package org.mca1;
public class Logging {
    public void beforeAdvice() {
        System.out.println("Going to setup student profile.");
    }
}
```

Student.java

```
package org.mca1;

public class Student {
    int age;
    String name;

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

StudentTest.java

```
package org.mca1;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import java.io.*;
public class StudentTest {
    private static ApplicationContext context;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        context = new ClassPathXmlApplicationContext("appctx.xml");
        Student student = (Student) context.getBean("student");
        String abc=student.getName();
    }
}
```

```

        System.out.println("Name is:"+abc);
        int ag=student.getAge();
        System.out.println("Age is:"+ag);
    }
}

```

In src/main/resources

appctx.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:context="http://www.springframework.org/schema/context"
        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/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd">

    <aop:config>
        <aop:aspect id = "log" ref = "logging">
            <aop:pointcut id = "selectAll"
                expression = "execution(* org.mca1.Student.getName(..))"/>

            <aop:before pointcut-ref="selectAll" method="beforeAdvice"/>
        </aop:aspect>
    </aop:config>
    <bean id = "student" class = "org.mca1.Student">
        <property name = "name" value = "Zara" />
        <property name = "age" value = "11"/>
    </bean>

    <!-- Definition for logging aspect -->
    <bean id = "logging" class = "org.mca1.Logging"/>
</beans>

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.mca1</groupId>
    <artifactId>org.mca1</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

```

```

<name>AopStudent</name>
<url>http://maven.apache.org</url>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
  <dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.2.19</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>1.9.6</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
  <dependency>

```

```

    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>1.9.6</version>
  </dependency>

```

```

</dependencies>
</project>

```

ecl - org.mca1/pom.xml - Eclipse IDE

File Edit Source Navigate Search Project SQL Editor Run Database Window Help

Package Explorer

- org.mca1
 - src/main/java
 - org.mca1
 - Logging.java
 - Student.java
 - StudentTest.java
 - src/main/resources
 - appctx.xml
 - src/test/java
 - src/test/resources
 - JRE System Library [J2SE-1.5]
 - Maven Dependencies
 - src
 - target
 - pom.xml

Student.java Logging.java appctx.xml org.mca1/pom.xml StudentTest.java

```

42 <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
43 <dependency>
44   <groupId>org.springframework</groupId>
45   <artifactId>spring-jdbc</artifactId>
46   <version>5.2.8.RELEASE</version>
47 </dependency>
48 <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
49 <dependency>
50   <groupId>org.springframework</groupId>
51   <artifactId>spring-aop</artifactId>
52   <version>5.2.8.RELEASE</version>
53 </dependency>
54
55 <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
56 <dependency>
57   <groupId>org.aspectj</groupId>
58   <artifactId>aspectjrt</artifactId>
59   <version>1.9.6</version>
60 </dependency>
61 <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
62 <dependency>
63   <groupId>org.aspectj</groupId>
64   <artifactId>aspectjweaver</artifactId>
65   <version>1.9.6</version>
66 </dependency>
67
68
69
70 </dependencies>
71 </project>
72
73
74

```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Problems Javadoc Declaration Console

```

<terminated> StudentTest (2) [Java Application] C:\Users\ADMIN\p2\pool\plugins\org.eclipse.justj.openj
Going to setup student profile.
Name is:Zara
Age is:11

```

project/dependencies/dependency/groupid/#text

Writeable

Program no.2 Write a program to demonstrate Spring AOP – after advice.

Logging.java

```
package org.mca2;

public class Logging {
    /**
     * This is the method which I would like to execute
     * after a selected method execution.
     */
    public void afterAdvice(){
        System.out.println("Student profile setup complete.");
    }
}
```

Student.java

```
package org.mca2;
public class Student {
    int age;
    String name;

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }
}
```

```

    }

    public void setName(String name) {
        this.name = name;
    }
}

```

StudentTest.java

```

package org.mca2;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import java.io.*;
public class StudentTest {
    private static ApplicationContext context;
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        context = new ClassPathXmlApplicationContext("appctx.xml");
        Student student = (Student) context.getBean("student");
        String abc=student.getName();
        System.out.println("Name is:"+abc);
        int ag=student.getAge();
        System.out.println("Age is:"+ag);
    }
}

```

In src/main/resources

appctx.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:context="http://www.springframework.org/schema/context"
    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/context
http://www.springframework.org/schema/context/spring-context-4.3.xsd">
```

```
<aop:config>
    <aop:aspect id = "log" ref = "logging">
        <aop:pointcut id = "selectAll"
            expression = "execution(* org.mca2.Student.getName(..))"/>

        <aop:after pointcut-ref="selectAll" method="afterAdvice"/>
    </aop:aspect>
</aop:config>
<bean id = "student" class = "org.mca2.Student">
    <property name = "name" value = "Zara" />
    <property name = "age" value = "11"/>
</bean>

<!-- Definition for logging aspect -->
<bean id = "logging" class = "org.mca2.Logging"/>
</beans>
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.mca2</groupId>
    <artifactId>org.mca2</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>AopStudent</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>5.2.8.RELEASE</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
```

```

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.8.RELEASE</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.2.19</version>
</dependency>

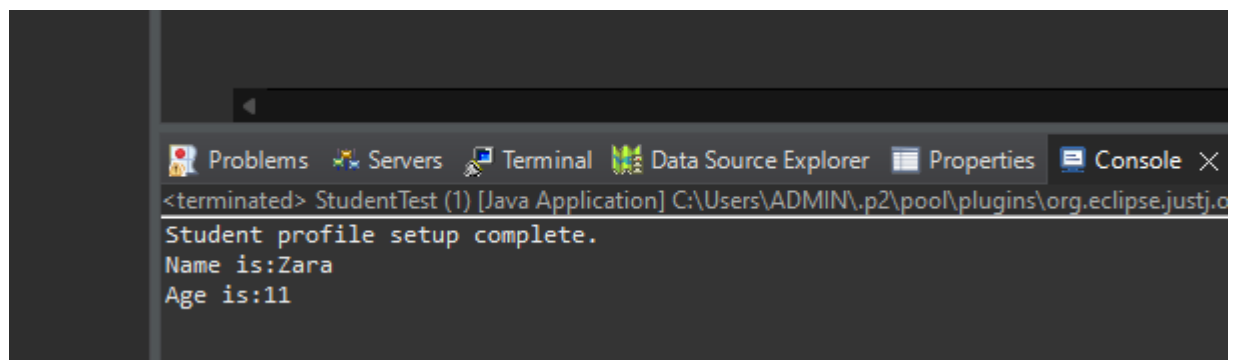
<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.2.8.RELEASE</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>
    <version>5.2.8.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>1.9.6</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>1.9.6</version>
</dependency>

</dependencies>

</project>

```



Program no. 3 Write a program to demonstrate Spring AOP – after returning advice.

Student.java:

```
package org.mca;

public class Student {
    private Integer age;
    private String name;
    public void setAge(Integer age) {
        this.age = age;
    }
    public Integer getAge() {
        System.out.println("Age : " + age );
        return age;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        System.out.println("Name : " + name );
        return name;
    }
    public void printThrowException(){
        System.out.println("Exception raised");
        throw new IllegalArgumentException();
    }
}
```

Logging.java:

```
package org.mca;

public class Logging {
    public void afterReturningAdvice(Object retVal){
        System.out.println("Returning:" + retVal.toString() );
    }
}
```

Beans.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"
    xsi:schemaLocation = "http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

    <aop:config>
        <aop:aspect id = "log" ref = "logging">
            <aop:pointcut id = "selectAll"
```

```

        expression = "execution(* org.mca.*.*(..))"/>
        <aop:after-returning pointcut-ref = "selectAll"
            method = "afterReturningAdvice" returning = "retVal"/>
    </aop:aspect>
</aop:config>

<!-- Definition for student bean -->
<bean id = "student" class = " org.mca.Student">
    <property name = "name" value = "Zara" />
    <property name = "age" value = "11"/>
</bean>

<!-- Definition for logging aspect -->
<bean id = "logging" class = " org.mca.Logging"/>
</beans>

```

MainApp.java:

```

package org.mca;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

```

```

public class MainApp {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext context = new
ClassPathXmlApplicationContext("Beans.xml");

        Student student = (Student) context.getBean("student");
        student.getName();
        student.getAge();
    }

}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.tutorialspoint</groupId>
    <artifactId>AfterReturningAdvice</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>AopStudent</name>
    <url>http://maven.apache.org</url>

```

```

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>3.8.1</version>
    <scope>test</scope>
  </dependency>

  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-core</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
  <dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <version>42.2.19</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
  <dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>
    <version>5.2.8.RELEASE</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
  <dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>1.9.6</version>
  </dependency>

  <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
  <dependency>
    <groupId>org.aspectj</groupId>

```

```

        <artifactId>aspectjweaver</artifactId>
        <version>1.9.6</version>
    </dependency>

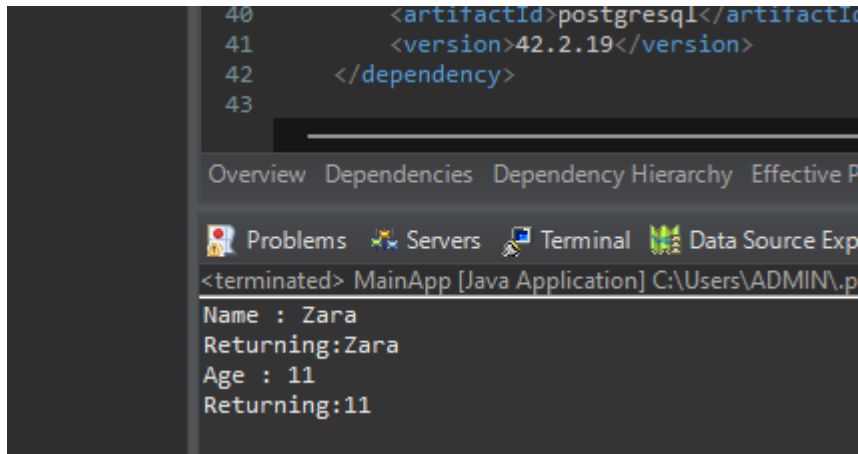
```

```

    </dependencies>
</project>

```

Output



Program no 4 Write a program to demonstrate Spring AOP –around advice.

Student.java

package org.viva2;

```

public class Student {
    private Integer age;
    private String name;
    public void setAge(Integer age) {
        this.age = age;
    }
    public Integer getAge() {
        System.out.println("Age : " + age );
        return age;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        System.out.println("Name : " + name );
        return name;
    }
    public void printThrowException(){
        System.out.println("Exception raised");
        throw new IllegalArgumentException();
    }
}

```

```
}
```

Logging.java

```
package org.viva2;
import org.aspectj.lang.ProceedingJoinPoint;
public class Logging {
    /**
     * This is the method which I would like to execute
     * around a selected method execution.
     */
    public String aroundAdvice(ProceedingJoinPoint jp) throws Throwable{
        System.out.println("Around advice");
        Object[] args = jp.getArgs();
        if(args.length>0){
            System.out.print("Arguments passed: " );
            for (int i = 0; i < args.length; i++) {
                System.out.print("arg "+(i+1)+" "+args[i]);
            }
        }
        Object result = jp.proceed(args);
        System.out.println("Returning " + result);
        return result.toString();
    }
}
```

Appctx.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"
    xsi:schemaLocation = "http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

    <aop:config>
        <aop:aspect id = "log" ref = "logging">
            <aop:pointcut id = "selectName"
                expression = "execution(* org.viva2.Student.getName(..))"/>
            <aop:around pointcut-ref = "selectName" method = "aroundAdvice"/>
        </aop:aspect>
    </aop:config>

    <!-- Definition for student bean -->
    <bean id = "student" class = "org.viva2.Student">
        <property name = "name" value = "Sanjana" />
        <property name = "age" value = "21"/>
    </bean>

    <!-- Definition for logging aspect -->
    <bean id = "logging" class = "org.viva2.Logging"/>
</beans>
```

StudentTest.java

```
package org.viva2;
```

```

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class StudentTest {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ApplicationContext context = new ClassPathXmlApplicationContext("Appctx.xml");

        Student student = (Student) context.getBean("student");
        student.getName();
    }
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.viva2</groupId>
    <artifactId>org.viva2</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>AopStudent</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>5.2.8.RELEASE</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.2.8.RELEASE</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
        <dependency>
            <groupId>org.postgresql</groupId>
            <artifactId>postgresql</artifactId>
            <version>42.2.19</version>

```

```

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.2.8.RELEASE</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-aop</artifactId>
    <version>5.2.8.RELEASE</version>
</dependency>

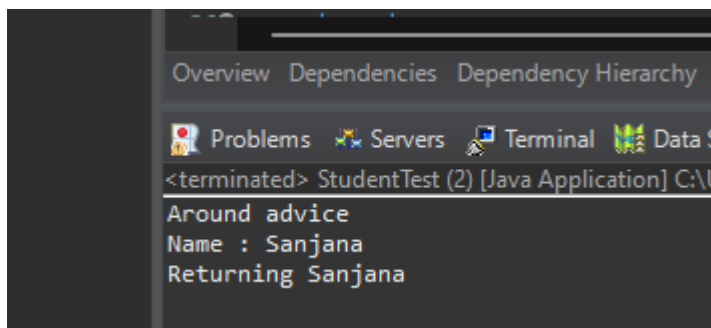
<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjrt</artifactId>
    <version>1.9.6</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>1.9.6</version>
</dependency>

```

```
</dependencies>
```

```
</project>
```

Output



Program no. 5 Write a program to demonstrate Spring AOP –After Throwing advice.

Student.java:

```
package org.viva4;
```

```

public class Student {
    private Integer age;
    private String name;
    public Integer getAge() {
        return age;
    }
}

```

```

    }
    public void setAge(Integer age) {
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void printThrowException(){
        System.out.println("Exception raised");
        throw new IllegalArgumentException();
    }
}

```

Logging.java:

```
package org.viva4;
```

```
import org.aspectj.lang.annotation.Aspect;
```

```
import org.aspectj.lang.JoinPoint;
```

```
import org.aspectj.lang.annotation.AfterThrowing;
```

```
@Aspect
```

```
public class Logging {
```

```
    public void afterThrowingAdvice(JoinPoint jp, Throwable error){
        System.out.println("Method Signature: " + jp.getSignature());
        System.out.println("Exception: "+error);
    }
}

```

Beans.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"
    xsi:schemaLocation = "http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop-3.0.xsd ">

```

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

```
<!-- Definition for student bean -->
```

```

<bean id = "student" class = "org.viva4.Student">
    <property name = "name" value = "Zara" />
    <property name = "age" value = "11"/>
</bean>

```

```
<!-- Definition for logging aspect -->
```

```
<bean id = "logging" class = "org.viva4.Logging"/>
```

```
</beans>
```

MainApp.java:

```
package org.viva4;
```



```

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("Beans.xml");

        Student student = (Student) context.getBean("student");
        student.printThrowException();
    }
}

```

Pom.xml

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
        https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.tutorialspoint</groupId>
    <artifactId>AfterThrowing</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>AopStudent</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>5.2.8.RELEASE</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.2.8.RELEASE</version>
        </dependency>

        <!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
        <dependency>

```

```

        <groupId>org.postgresql</groupId>
        <artifactId>postgresql</artifactId>
        <version>42.2.19</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>5.2.8.RELEASE</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aop</artifactId>
        <version>5.2.8.RELEASE</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjrt -->
    <dependency>
        <groupId>org.aspectj</groupId>
        <artifactId>aspectjrt</artifactId>
        <version>1.9.6</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.aspectj/aspectjweaver -->
    <dependency>
        <groupId>org.aspectj</groupId>
        <artifactId>aspectjweaver</artifactId>
        <version>1.9.6</version>
    </dependency>

</dependencies>
</project>

```

Output:

```

Exception raised
Exception in thread "main" java.lang.IllegalArgumentException
    at org.viva4.Student.printThrowException(Student.java:20)
    at org.viva4.MainApp.main(MainApp.java:13)

```

Program 6 Write a program to demonstrate Spring AOP – pointcuts.
MyService.java
package com.example.aopdemo;

```

public class MyService {
    public void performTask() {
        System.out.println("Executing the task in MyService.");
    }
}

```

```

    }
}
LoggingAspect.java
package com.example.aopdemo;

import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;

@Aspect
public class LoggingAspect {

    // Define a pointcut expression
    @Before("execution(* com.example.aopdemo.MyService.performTask(..))")
    public void logBefore() {
        System.out.println("LoggingAspect: Before executing performTask.");
    }
}
applicationContext.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"
    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.xsd">

    <!-- Enable AspectJ auto proxy -->
    <aop:aspectj-autoproxy/>

    <!-- Register beans -->
    <bean id="myService" class="com.example.aopdemo.MyService"/>
    <bean id="loggingAspect" class="com.example.aopdemo.LoggingAspect"/>

</beans>

MainApp.java
package com.example.aopdemo;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        // Load the Spring context
        ApplicationContext context = new
        ClassPathXmlApplicationContext("applicationContext.xml");

```

```

// Retrieve the service bean
MyService myService = context.getBean("myService", MyService.class);

// Call the method to see AOP in action
myService.performTask();

// Close the context
((ClassPathXmlApplicationContext) context).close();
}
}
Pom.xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>SpringAOPDemo</artifactId>
  <version>1.0-SNAPSHOT</version>

  <dependencies>
    <!-- Spring Context for AOP -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>5.3.30</version>
    </dependency>

    <!-- Spring AOP -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-aop</artifactId>
      <version>5.3.30</version>
    </dependency>

    <!-- AspectJ for annotations -->
    <dependency>
      <groupId>org.aspectj</groupId>
      <artifactId>aspectjweaver</artifactId>
      <version>1.9.19</version>
    </dependency>
  </dependencies>
</project>

```

Output

```
<terminated> MainApp (2) [Java Application] C:\Users\ADMIN\p2\pool\p  
LoggingAspect: Before executing performTask.  
Executing the task in MyService.
```