

# PRACTICAL 8(F) – SPRING AOP (POINTCUTS)

## AIM

To demonstrate the use of Pointcuts in Spring AOP.

## THEORY

A Pointcut is an expression that defines where an advice should be applied. It separates the definition of cross-cutting concerns from business logic. Pointcuts can be reused across multiple advices.

## SOURCE CODE – Target Class (BusinessLogic.java)

```
package aopdemo;

public class BusinessLogic {
    public void process() {
        System.out.println("Processing business logic");
    }

    public void validate() {
        System.out.println("Validating business data");
    }
}
```

## SOURCE CODE – Aspect Class (PointcutAspect.java)

```
package aopdemo;

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

@Aspect
public class PointcutAspect {

    @Pointcut("execution(* aopdemo.BusinessLogic.*(..))")
    public void allBusinessMethods() {}

    @Before("allBusinessMethods()")
    public void beforeAdvice() {
        System.out.println("Before advice using Pointcut");
    }
}
```

## SOURCE CODE – 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">
<aop:aspectj-autoproxy/>
<bean id="business" class="aopdemo.BusinessLogic"/>
<bean class="aopdemo.PointcutAspect"/>
</beans>

```

## SOURCE CODE – MainApp.java

```

package aopdemo;

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

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

        BusinessLogic obj = (BusinessLogic) context.getBean("business");
        obj.process();
        obj.validate();
    }
}

```

## NETBEANS 7 – EXECUTION STEPS

1. Use the same Spring AOP project and libraries from Practical 8(A)
2. Create PointcutAspect.java
3. Update BusinessLogic.java with multiple methods
4. Update applicationContext.xml
5. Run MainApp.java using Shift + F6

## OUTPUT

Before advice using Pointcut Processing business logic Before advice using Pointcut Validating business data

## RESULT

Thus, Pointcuts were successfully implemented in Spring AOP.