PRACTICAL 8(A) – SPRING AOP (BEFORE ADVICE)

GENERAL SETUP FOR SPRING AOP PRACTICALS (PRACTICAL 8)

This setup is common for all Spring AOP practicals (8A–8F). Spring AOP requires additional libraries apart from Spring Core.

STEP 1: SOFTWARE REQUIREMENTS

• JDK: Version 7 or 8
• NetBeans IDE 7.x
• Spring Framework JARs
• Spring AOP & AspectJ libraries

STEP 2: REQUIRED JAR FILES

Spring Core:
• spring-core.jar
• spring-beans.jar
• spring-context.jar
• spring-expression.jar
• commons-logging.jar

Spring AOP:
• spring-aop.jar
• aspectjweaver.jar
• aspectjrt.jar

STEP 3: CREATE JAVA APPLICATION PROJECT

1. Open NetBeans IDE 7
2. File → New Project → Java → Java Application
3. Project Name: SpringAOPDemo
4. Uncheck 'Create Main Class'
5. Click Finish

STEP 4: ADD EXTERNAL LIBRARIES

1. Right-click Project → Properties
2. Select Libraries
3. Click Add JAR/Folder
4. Add all Spring Core and Spring AOP JAR files
5. Click OK

PRACTICAL 8(A): SPRING AOP – BEFORE ADVICE

AIM

To demonstrate Spring AOP – Before Advice.

THEORY

Aspect Oriented Programming (AOP) is used to separate cross-cutting concerns such as logging, security, and transaction management. A Before Advice executes before the actual method execution.

SOURCE CODE – Target Class (BusinessLogic.java)

```
package aopdemo;

public class BusinessLogic {
    public void process() {
        System.out.println("Business logic executed");
    }
}
```

SOURCE CODE – Aspect Class (LoggingAspect.java)

```
package aopdemo;

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

@Aspect
public class LoggingAspect {

    @Before("execution(* aopdemo.BusinessLogic.process(..))")
    public void beforeAdvice() {
        System.out.println("Before advice executed");
    }
}
```

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.LoggingAspect"/>
</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();
    }
}
```

EXECUTION STEPS – PRACTICAL 8(A)

1. Create package aopdemo
2. Create BusinessLogic.java, LoggingAspect.java, MainApp.java
3. Create applicationContext.xml in Source Packages
4. Paste respective code
5. Run MainApp.java using Shift + F6

OUTPUT

Before advice executed Business logic executed

RESULT

Thus, Spring AOP Before Advice was successfully implemented.