PRACTICAL 8(D) – SPRING AOP (AFTER RETURNING ADVICE)

AIM

To demonstrate Spring AOP – After Returning Advice.

THEORY

After Returning Advice executes only after a method successfully returns a value. It is mainly used to perform operations such as logging, validation, or processing the returned data.

SOURCE CODE – Target Class (BusinessLogic.java)

```
package aopdemo;

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

SOURCE CODE – Aspect Class (ReturnAspect.java)

```
package aopdemo;

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

@Aspect
public class ReturnAspect {

    @AfterReturning(
        pointcut = "execution(* aopdemo.BusinessLogic.process(..))",
        returning = "result")
    public void afterReturningAdvice(Object result) {
        System.out.println("After Returning advice executed");
        System.out.println("Returned Value: " + result);
    }
}
```

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

NETBEANS 7 – EXECUTION STEPS

1. Use the same Spring AOP project and libraries from Practical 8(A)
2. Create ReturnAspect.java in aopdemo package
3. Update BusinessLogic.java to return a value
4. Update applicationContext.xml
5. Run MainApp.java using Shift + F6

OUTPUT

```
Business logic executed After Returning advice executed Returned Value: SUCCESS
```

RESULT

Thus, Spring AOP After Returning Advice was successfully implemented.