**ADVANCED JAVA**

**ASSIGNMENT- 6**

**(PROGRAMS ON SPRING-HIBERNATE FRAMEWORK AND JAVA BEANS)**

**NAME: JIGYASU SINGH CHOUHAN**

**PRN: 1132220615**

**Q1) write a simple java program using spring**

**Myspringprogram.java**

Import org.springframework.context.applicationcontext;

Import org.springframework.context.support.classpathxmlapplicationcontext;

Public class myspringprogram {

public static void main(string[] args) {

applicationcontext context = new classpathxmlapplicationcontext("applicationcontext.xml");

myservice myservice = (myservice) context.getbean("myservice");

string result = myservice.sayhello("john");

system.out.println(result);

}

}

**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"

xsi:schemalocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="myservice" class="com.example.myservice">

<property name="message" value="hello, %s!" />

</bean>

</beans>

**Output:**



**Q2)write a simple java program using explaining spring**

**Myapp.java**

Import org.springframework.context.applicationcontext;

Import org.springframework.context.support.classpathxmlapplicationcontext;

Public class myapp {

public static void main(string[] args) {

applicationcontext context = new classpathxmlapplicationcontext("applicationcontext.xml");

myservice myservice = (myservice) context.getbean("myservice");

myservice.sayhello();

}

}

Public interface myservice {

void sayhello();

}

Public class myserviceimpl implements myservice {

@override

public void sayhello() {

system.out.println("hello, world!");

}

}

**Applicationcontext.xml**

<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=”greetingservice” class=”greetingserviceimpl”>

<property name=”greeting” value=”hello, spring!”/>

</bean>

</beans>

**Output:**



**Q3) write a simple java program using explaining spring aop**

***MyApp.Java***

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MyApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

MyService myService = (MyService) context.getBean("myService");

myService.sayHello();

}

}

public interface MyService {

void sayHello();

}

public class MyServiceImpl implements MyService {

@Override

public void sayHello() {

System.out.println("Hello, world!");

}

}

public class LoggingAspect {

public void beforeAdvice() {

System.out.println("Before advice: Method is about to be executed.");

}

}

**applicationContext .Xml**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:aop="http://www.springframework.org/schema/aop"

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

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd">

<bean id="myService" class="MyServiceImpl" />

<bean id="loggingAspect" class="LoggingAspect" />

<aop:config>

<aop:aspect ref="loggingAspect">

<aop:before pointcut="execution(\* MyService.\*(..))" method="beforeAdvice" />

</aop:aspect>

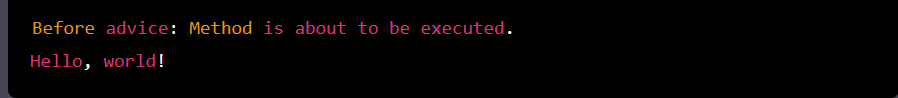
</aop:config>

</beans>

**Without AOP:**

****

**With AOP:**

****

**Q4) write a simple java program using java bean**

**Person..java**

public class Person {

private String name;

private int age;

private boolean married;

public Person() {}

public Person(String name, int age, boolean married) {

this.name = name;

this.age = age;

this.married = married;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

public boolean isMarried() {

return married;

}

public void setMarried(boolean married) {

this.married = married;

}

}

public class MyApp {

public static void main(String[] args) {

Person person = new Person();

person.setName("Jigyasu");

person.setAge(21);

person.setMarried(false);

System.out.println("Name: " + person.getName());

System.out.println("Age: " + person.getAge());

System.out.println("Married: " + person.isMarried());

}

}

**Output:**

