Name:____StudentID:____________________

3/3

Quiz W2D4: Spring DI & AOP

CS544 Enterprise Architecture

What is the output? **Important hints**: When making a subclass proxy it no longer seems to call the superclass constructor… Not sure how it's doing that.

```
<?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-4.2.xsd
       http://www.springframework.org/schema/aop
       http://www.springframework.org/schema/aop/spring-aop-4.2.xsd">
    <bean id="mySuper" class="cs544.w2d4q.MySuper" abstract="true">
        <property name="text" value="Test" />
    </bean>
    <bean id="myClass" class="cs544.w2d4q.MyClass" parent="mySuper"
          lazy-init="false" >
        <constructor-arg value="Thing" />
    </bean>
    <bean id="traceAdvice" class="cs544.w2d4q.TraceAdvice"
          init-method="start">
        <property name="text" value="Injected" />
    </bean>
    <aop:aspectj-autoproxy proxy-target-class="true" />
</beans>
```

```
public class App {
    public static void main(String[] args) {
        ConfigurableApplicationContext context = new
            ClassPathXmlApplicationContext("springconfig.xml");
        System.out.println("Testing Spring Startup");
        MyClass mc = context.getBean("myClass", MyClass.class);
        mc.sayHello();
    }
}
```

```
public abstract class MySuper {
    private String text;
    public void setText(String text) { this.text = text; }
    public String getText() { return text; }
}
```

```
public class MyClass extends MySuper {
    public MyClass(String text) {
        setText(text);
        System.out.println("MyClass Constructor");
    }
    public void sayHello() {
        System.out.println("This is a " + getText());
    }
}
```

Name:____StudentID:____________________

```java
@Aspect
public class TraceAdvice {
    private String text;
    public TraceAdvice() {
        System.out.println("TraceAdvice Constructor - text: " + text);
    }
    public void setText(String text) {
        this.text = text;
    }
    public void start() {
        System.out.println("TraceAdvice start method - text: " + text);
    }
    @Before("execution(* cs544.w2d4q.*.*(..))")
    public void beforeTrace(JoinPoint jp) {
        System.out.println("Before: " + jp.getSignature().getName());
    }
}
```

Output:

```
1 MyClass Constructor
2 TraceAdvice Constructor - text: null
3 TraceAdvice start method - text: Injected
4 Testing Spring Startup
5 Before: sayHello
6 This is a Test
```

6/6 → 3/3

Thing I cannot proxy

- local calls
- when advice uses 'target'
- Constructor
- static method