Code Explanation

This code is to show how Spring (a framework in java) uses proxy classes for Aspect Oriented Programming(AOP). Spring uses a process called weaving to construct proxy classes. This code shows what proxy classes look like after the process of weaving(weaving LoggingAspect.java and student.java to create StudentProxy.java).

In this code, the Student class has name and ID as attributes and it contains all setters and getters methods. To implement a logging method in the getName method in student, first, we created a logging method as getNameAdvice in a class named LoggingAspect. In spring, we have to instantiate objects it's working on, i.e we have to get the objects from spring to access methods of the class they belong to. We created our classes in such a way that we get the proxy object from the spring service class. In Aop.java, which is the main file of all where student object has been created from the Spring service and getName method has been called.

Spring automatically creates the proxy objects and returns them instead of the actual object. StudentProxy class has a new method (getName) that calls both logging method (getNameAdvice) and the actual getName method. So, it returns the actual method along with the logging method whenever the getName is called.

How to run the code--

1) javac *.java
2) java Aop