XSTREAM - FIRST APPLICATION

Before going into the details of the XStream library, let us see an application in action. In this example, we've created Student and Address classes. We will create a student object and then serialize it to an XML String. Then de-serialize the same XML string to obtain the student object back.

Create a java class file named XStreamTester in **C:\>XStream_WORKSPACE**.

File: XStreamTester.java

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;

import javax.xml.transform.OutputKeys;
import javax.xml.transform.Source;
import javax.xml.transform.Transformer;

import javax.xml.transform.sax.SAXSource;
import javax.xml.transform.sax.SAXTransformerFactory;
import javax.xml.transform.stream.StreamResult;

import org.xml.sax.InputSource;

import com.thoughtworks.xstream.XStream;
import com.thoughtworks.xstream.io.xml.StaxDriver;

public class XStreamTester {
   public static void main(String args[]){

      XStreamTester tester = new XStreamTester();
      XStream xstream = new XStream(new StaxDriver());

      Student student = tester.getStudentDetails();

      //Object to XML Conversion
      String xml = xstream.toXML(student);
      System.out.println(formatXml(xml));

      //XML to Object Conversion
      Student student1 = (Student)xstream.fromXML(xml);
      System.out.println(student1);
   }

   private Student getStudentDetails(){

      Student student = new Student();

      student.setFirstName("Mahesh");
      student.setLastName("Parashar");
      student.setRollNo(1);
      student.setClassName("1st");

      Address address = new Address();

      address.setArea("H.No. 16/3, Preet Vihar.");
      address.setCity("Delhi");
      address.setState("Delhi");
      address.setCountry("India");
      address.setPincode(110012);

      student.setAddress(address);

      return student;
   }
```

```
   public static String formatXml(String xml){

      try{
         Transformer serializer = SAXTransformerFactory.newInstance().newTransformer();

         serializer.setOutputProperty(OutputKeys.INDENT, "yes");
         serializer.setOutputProperty("{http://xml.apache.org/xslt}indent-amount", "2");

         Source xmlSource = new SAXSource(new InputSource(new
ByteArrayInputStream(xml.getBytes())));
         StreamResult res =  new StreamResult(new ByteArrayOutputStream());

         serializer.transform(xmlSource, res);

         return new String(((ByteArrayOutputStream)res.getOutputStream()).toByteArray());

      }catch(Exception e){
         return xml;
      }
   }
}

class Student {

   private int rollNo;

   private String firstName;
   private String lastName;
   private String className;

   private Address address;

   public String getFirstName() {
      return firstName;
   }

   public void setFirstName(String firstName) {
      this.firstName = firstName;
   }

   public String getLastName() {
      return lastName;
   }

   public void setLastName(String lastName) {
      this.lastName = lastName;
   }

   public int getRollNo() {
      return rollNo;
   }

   public void setRollNo(int rollNo) {
      this.rollNo = rollNo;
   }

   public String getClassName() {
      return className;
   }

   public void setClassName(String className) {
      this.className = className;
   }

   public Address getAddress() {
      return address;
   }
```

```
   public void setAddress(Address address) {
      this.address = address;
   }

   public String toString(){

      StringBuilder stringBuilder = new StringBuilder();

      stringBuilder.append("Student [ ");
      stringBuilder.append("\nfirstName: ");
      stringBuilder.append(firstName);
      stringBuilder.append("\nlastName: ");
      stringBuilder.append(lastName);
      stringBuilder.append("\nrollNo: ");
      stringBuilder.append(rollNo);
      stringBuilder.append("\nclassName: ");
      stringBuilder.append(className);
      stringBuilder.append("\naddress: ");
      stringBuilder.append(address);
      stringBuilder.append(" ]");

      return stringBuilder.toString();
   }
}

class Address {

   private String area;
   private String city;
   private String state;
   private String country;

   private int pincode;

   public String getArea() {
      return area;
   }

   public void setArea(String area) {
      this.area = area;
   }

   public String getCity() {
      return city;
   }

   public void setCity(String city) {
      this.city = city;
   }

   public String getState() {
      return state;
   }

   public void setState(String state) {
      this.state = state;
   }

   public String getCountry() {
      return country;
   }

   public void setCountry(String country) {
      this.country = country;
   }

   public int getPincode() {
      return pincode;
   }
```

```
   public void setPincode(int pincode) {
      this.pincode = pincode;
   }

   public String toString(){

      StringBuilder stringBuilder = new StringBuilder();

      stringBuilder.append("\nAddress [ ");
      stringBuilder.append("\narea: ");
      stringBuilder.append(area);
      stringBuilder.append("\ncity: ");
      stringBuilder.append(city);
      stringBuilder.append("\nstate: ");
      stringBuilder.append(state);
      stringBuilder.append("\ncountry: ");
      stringBuilder.append(country);
      stringBuilder.append("\npincode: ");
      stringBuilder.append(pincode);
      stringBuilder.append(" ]");

      return stringBuilder.toString();
   }
}
```

Verify the Result

Compile the classes using **javac** compiler as follows:

```
C:\XStream_WORKSPACE>javac XStreamTester.java
```

Now run the XStreamTester to see the result:

```
C:\XStream_WORKSPACE>java XStreamTester
```

Verify the output as follows

```
<?xml version="1.0" encoding="UTF-8"?>
<Student>
   <firstName>Mahesh</firstName>
   <lastName>Parashar</lastName>
   <rollNo>1</rollNo>
   <className>1st</className>
   <address>
      <area>H.No. 16/3, Preet Vihar.</area>
      <city>Delhi</city>
      <state>Delhi</state>
      <country>India</country>
      <pincode>110012</pincode>
   </address>
</Student>

Student [
firstName: Mahesh
lastName: Parashar
rollNo: 1
className: 1st
address:
Address [
area: H.No. 16/3, Preet Vihar.
city: Delhi
state: Delhi
country: India
pincode: 110012 ] ]
```

Steps to Remember

Following are the important steps to be considered here.

Step 1: Create an XStream Object

Create an XStream object by passing it a StaxDriver. StaxDriver uses Stax pull parser *availablefromjava*6 and is a fast xml parser.

```
XStream xstream = new XStream(new StaxDriver());
```

Step 2: Serialize the Object to XML

Use toXML method to get the XML string representation of the object.

```
//Object to XML Conversion
String xml = xstream.toXML(student);
```

Step 3: De-serialize XML to Get the Object

Use fromXML method to get the object from the XML.

```
//XML to Object Conversion
Student student1 = (Student)xstream.fromXML(xml);
```