Persistence

An Introduction to XML and Serialization

Produced by: Dr. Siobhán Drohan
Maireád Meagher

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/

Topic List

- Introduction to XML:
  - XML versus HTML
  - Example of XML
  - XML does not “do” anything
- Object Serialization.
- ShopV4.0 – XML persistence.

XML versus HTML

- XML was designed to describe data, with a focus on what the data is. XML is about carrying information.
- HTML was designed to display data, with a focus on how the data looks. HTML is about displaying information.
- XML is NOT a replacement for HTML.

Source: http://www.w3schools.com/xml/xml_whatis.asp

XML…

…stands for EXtensible Markup Language

…is a markup language much like HTML

…was designed to describe data, not to display data

…tags are not predefined. You must define your own tags

…is designed to be self-descriptive.

Source: http://www.w3schools.com/xml/xml_whatis.asp

Example of XML

- The following example is a note to Mairead, from Siobhan, stored as XML:

```
<note>
        <to>Mairead</to>
        <from>Siobhan</from>
        <heading>Reminder</heading>
        <body>Meeting at 10am today</body>
</note>
```

- The note above is quite self descriptive. It has sender and receiver information, it also has a heading and a message body.

Shop V3.0

Shop V3.0 has implemented the CRUD process, but when we close down our application, all the entered data is lost.

Shop V4.0 - We will use XML to make our data persistent. We will store our objects to XML files.

XML does not “do” anything

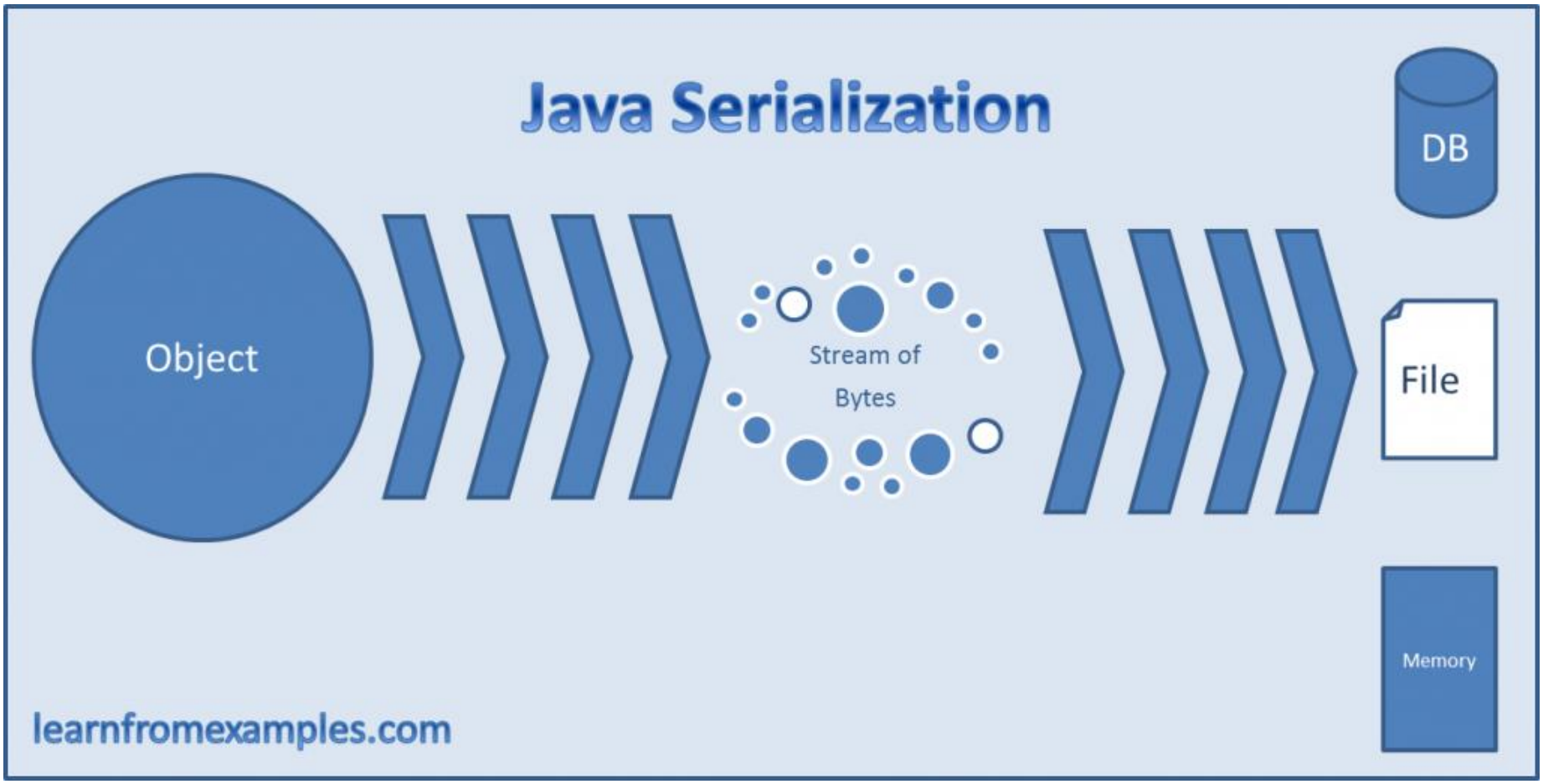
- XML is just information wrapped in tags. Someone must write a piece of software to send, receive or display it.

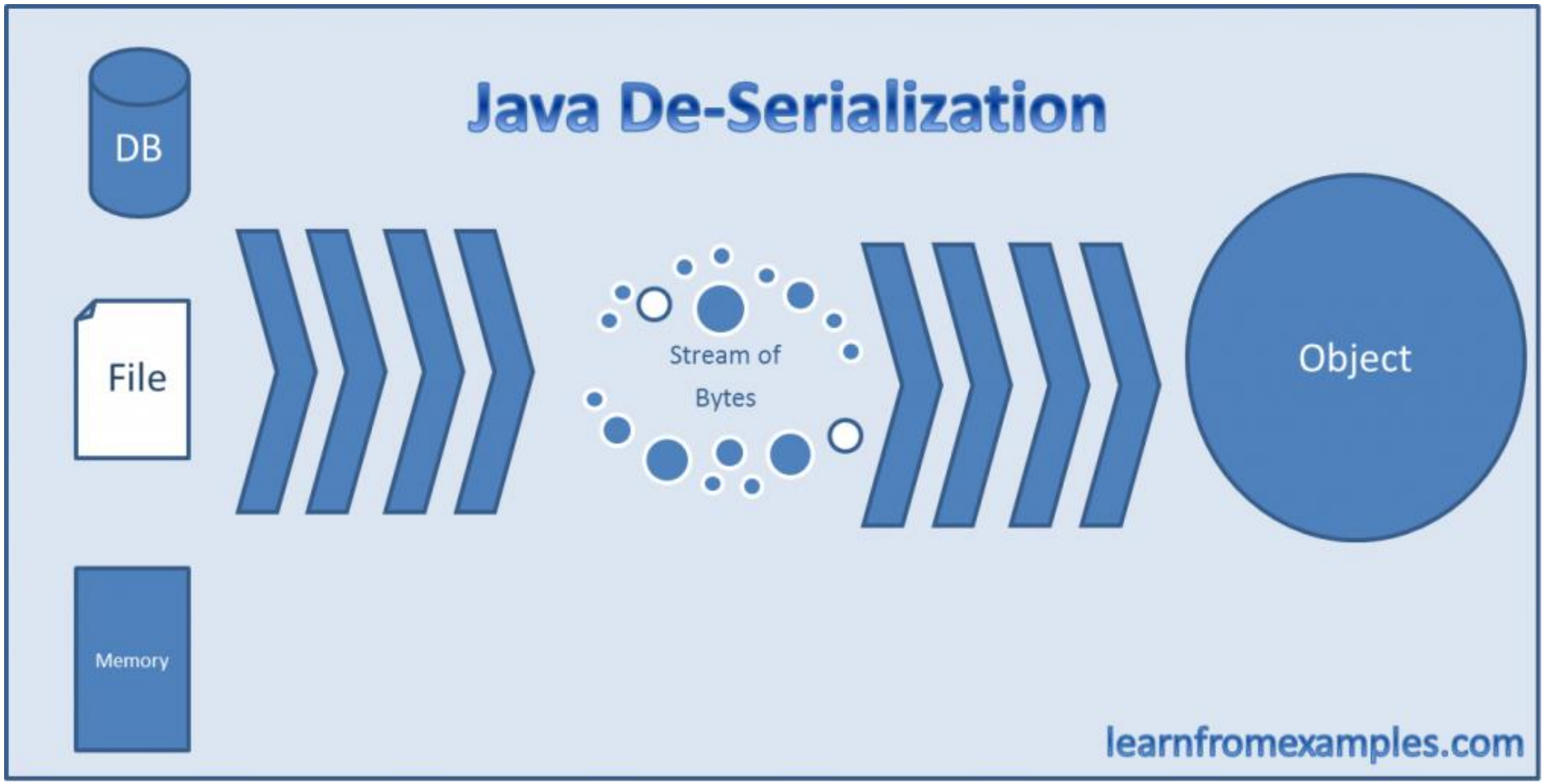
- We will write Java code to:
  - send our objects to an XML file on the hard disk.
  - read our objects from an XML file on the hard disk.

Source: http://www.w3schools.com/xml/xml_whatis.asp

Topic List

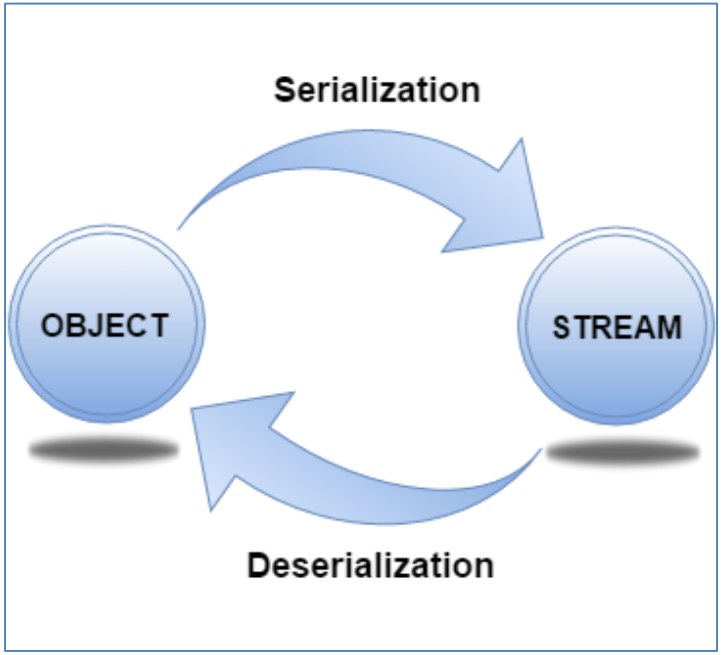
- Introduction to XML:
  - XML versus HTML
  - Example of XML
  - XML does not “do” anything
- Object Serialization.
- ShopV4.0 – XML persistence.


Java Serialization
DB
Object
Stream of
Bytes
File
Memory
learnfromexamples.com


Java De-Serialization
DB
File
Memory
Stream of
Bytes
Object
learnfromexamples.com

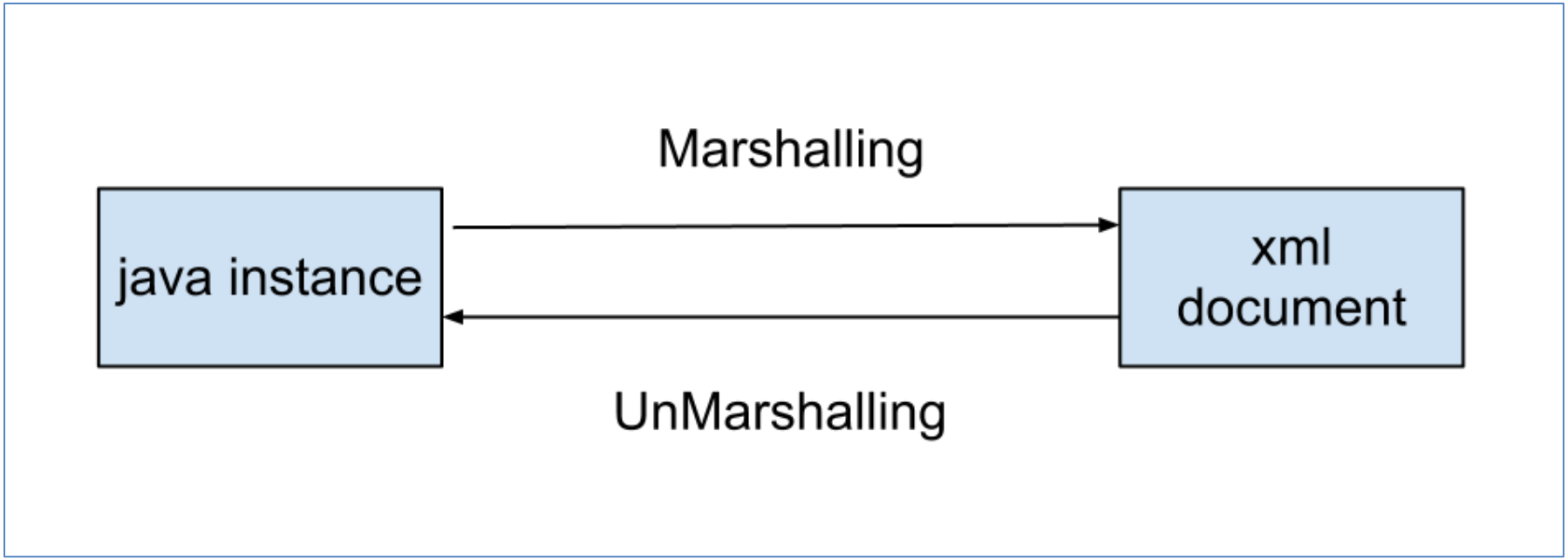
Object serialization

An object can be represented as a sequence of bytes that includes the object's data as well as information about the object's type and the types of data stored in the object.



Serialization process involves Marshalling and unMarshalling

Marshalling…the process of converting the objects & data into a stream.



UnMarshalling is the reverse process of converting the stream back to their original objects & data.

Topic List

- Introduction to XML:
  - XML versus HTML
  - Example of XML
  - XML does not “do” anything
- Object Serialization.
- ShopV4.0 – XML persistence.

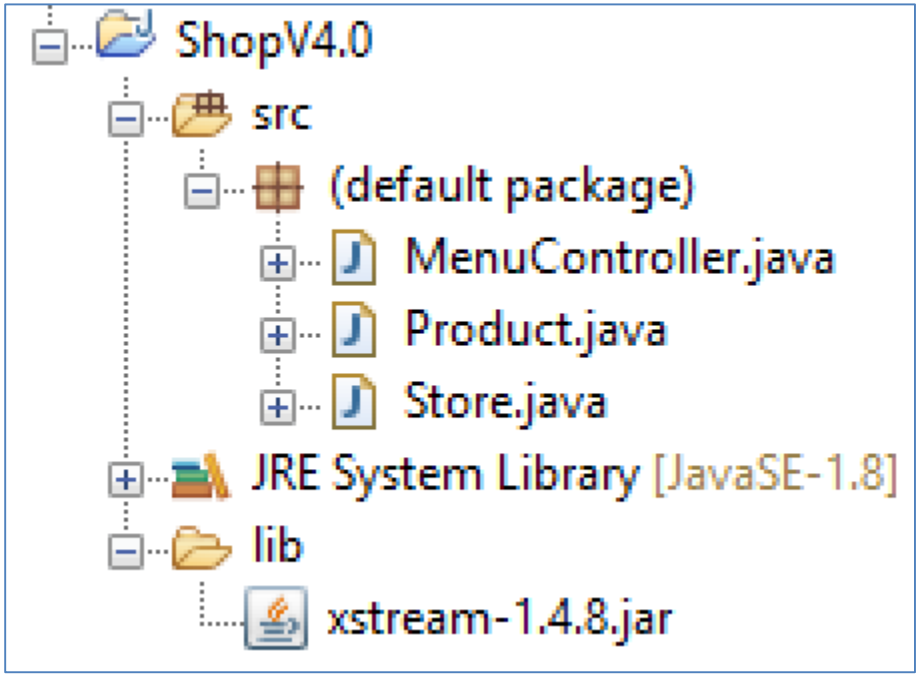
Shop V4.0 (using XML)

1. Download the **xstream-1.4.8.jar** component and add it to your Shop project.

2. Store Class – write the load(), save() methods.

3. MenuController Class - include extra load and save functionality to the menu.

Adding a component to the lib folder

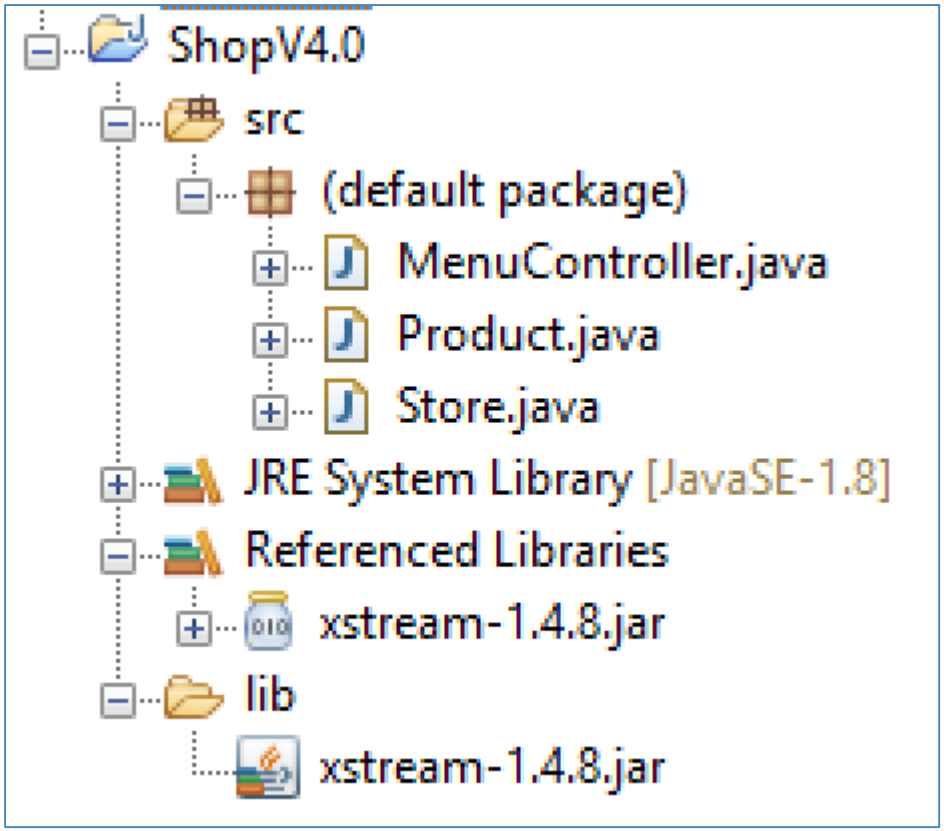


Download the xstream-1.4.8.jar component.

In Eclipse Package Explorer, create a new folder "lib“ (select "File->New->Folder“)

Drag the xstream-1.4.8.jar component into the lib folder (choose the option to copy the file).

Adding the component to your build path



Right click on the xstream-1.4.8.jar file and select:

"Build Path->Add to Build Path".

Shop V4.0 (using XML)

1. Download the **xstream-1.4.8.jar** component and add it to your Shop project.

2. Store Class – write the load(), save() methods.

3. MenuController Class - include extra load and save functionality to the menu.

Store.java

To use this code in another project, change:

- the **type** of object stored in the ArrayList.
- the name of the xml file
- The name of the ArrayList object.

```
@SuppressWarnings("unchecked")
public void load() throws Exception
{
    XStream xstream = new XStream(new DomDriver());
    ObjectInputStream is = xstream.createObjectInputStream
                                (new FileReader("products.xml"));
    products = (ArrayList<Product>) is.readObject();
    is.close();
}

public void save() throws Exception
{
    XStream xstream = new XStream(new DomDriver());
    ObjectOutputStream out = xstream.createObjectOutputStream
                                (new FileWriter("products.xml"));
    out.writeObject(products);
    out.close();
}
```

Store.java

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;

import com.thoughtworks.xstream.XStream;
import com.thoughtworks.xstream.io.xml.DomDriver;
```

Note: you will need to import these additional packages.

Shop V4.0 (using XML)

1. Download the **xstream-1.4.8.jar** component and add it to your Shop project.

2. Store Class – write the load(), save() methods.

3. MenuController Class - include extra load and save functionality to the menu.

MenuController class code

```
private int mainMenu()
{
    System.out.println("\fShop Menu");
    System.out.println("---------");
    System.out.println("  1) Add a Product");
    System.out.println("  2) List the Products");
    System.out.println("  3) Update a Product");
    System.out.println("  4) Remove Product (by index)");
    System.out.println("---------");
    System.out.println("  5) List the cheapest product");
    System.out.println("---------");
    System.out.println("  6) View store details");
    System.out.println("---------");
    System.out.println("  7) Save products (XML)");
    System.out.println("  8) Load products (XML)");
    System.out.println("  0) Exit");
    System.out.print("==>> ");
    int option = input.nextInt();
    return option;
}
```

```
Shop Menu
---------
  1) Add a Product
  2) List the Products
  3) Update a Product
  4) Remove Product (by index)
---------
  5) List the cheapest product
---------
  6) View store details
---------
  7) Save products (XML)
  8) Load products (XML)
  0) Exit
==>>
```

Add the Save and Load functionality to the menu system.

MenuController class code

```
case 7:     try {
               store.save();
            }
            catch (Exception e) {
               System.err.println("Error writing to file: " + e);
            }
            break;
case 8:     try {
               store.load();
            }
            catch (Exception e) {
                System.err.println("Error reading from file: " + e);
            }
            break;
```

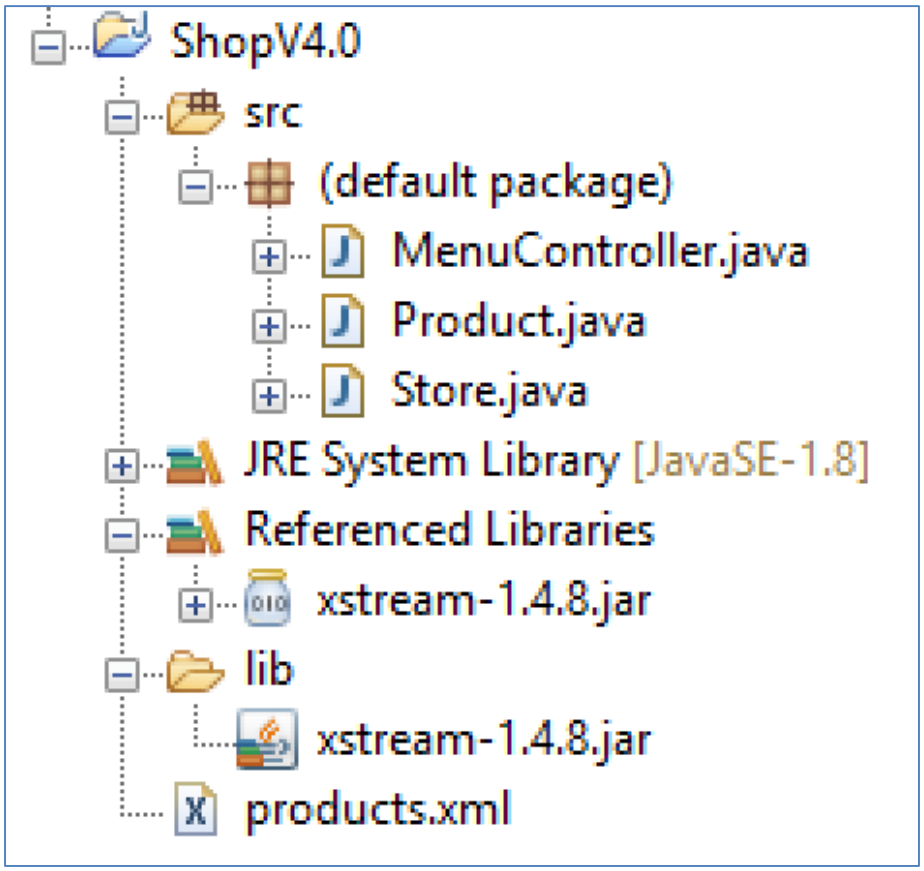
Add the Save and Load functionality to the menu system.

Inside a try/catch block, call the save method for option 7, and the load method for option 8.

```
□Shop Menu
---------
  1) Add a Product
  2) List the Products
  3) Update a Product
  4) Remove Product (by index)
---------
  5) List the cheapest product
---------
  6) View store details
---------
  7) Save products (XML)
  8) Load products (XML)
  0) Exit
==>> 8

Press any key to continue...
Error reading from file: java.io.FileNotFoundException: products.xml (The system cannot find the file specified)
```

Attempting to load a file that doesn't exist



The XML file that is created when the save option is selected from the menu; it can be found in your project directory.

```
<object-stream>
  <list>
    <Product>
      <productName>Coca Cola</productName>
      <productCode>1003</productCode>
      <unitCost>1.35</unitCost>
      <inCurrentProductLine>true</inCurrentProductLine>
    </Product>
    <Product>
      <productName>Fanta</productName>
      <productCode>1006</productCode>
      <unitCost>1.29</unitCost>
      <inCurrentProductLine>true</inCurrentProductLine>
    </Product>
  </list>
</object-stream>
```

Any Questions?

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/