Introduction to XML and Basic Operations

Lab Tasks:

Creating Your First XML Document

1. **Create an XML Document**:

o Open your IDE (NetBeans, Eclipse, etc.) and create a new **Java Project**. o Create a new file named books.xml in your project folder.

o Use the following example to create an XML document representing a simple list of books:

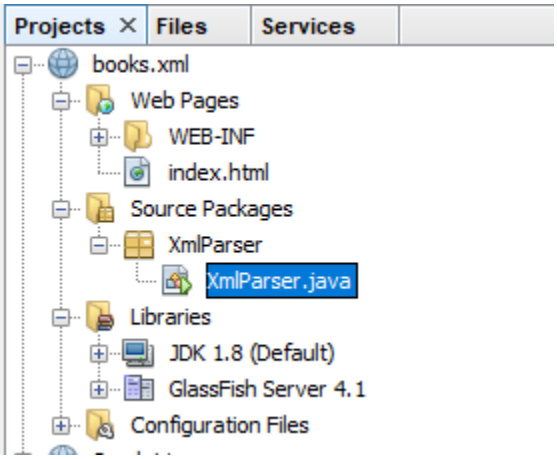
```
<?xml version="1.0" encoding="UTF-8"?>
 <library>
 <book>
 <title>The Great Gatsby</title>
 <author>F. Scott Fitzgerald</author>
 <year>1925</year>
 <genre>Fiction</genre>
 </book>
 <book>
 <title>To Kill a Mockingbird</title>
 <author>Harper Lee</author>
 <year>1960</year>
 <genre>Fiction</genre>
 </book>
 <book>
 <title>1984</title>
 <author>George Orwell</author>
 <year>1949</year>
 <genre>Dystopian</genre>
 </book>
</library>
```

Parsing XML in Java

Now that we have created an XML document, let's read and parse it using Java. We'll use **Java DOM (Document Object Model)** parsing for this task.

1. **Create a Java Class for XML Parsing**:

o Create a new Java class named **XmlParser.java** in your project.



o Add the following code to read and parse the books.xml file

```
package XmlParser;

import org.w3c.dom.*;

import javax.xml.parsers.*;


public class XmlParser {

    public static void main(String[] args) {


 try {

// Create a new DocumentBuilderFactory and DocumentBuilder

 DocumentBuilderFactory factory =  DocumentBuilderFactory.newInstance();

 DocumentBuilder builder = factory.newDocumentBuilder();

 // Parse the XML file

 Document document = builder.parse("books.xml");
```

```
// Normalize the document
document.getDocumentElement().normalize();
// Get the root element (library)
NodeList nodeList = document.getElementsByTagName("book");
// Loop through each book in the XML document
for (int i = 0; i < nodeList.getLength(); i++) { Node node = nodeList.item(i);

if (node.getNodeType() == Node.ELEMENT_NODE) { Element element = (Element) node;

// Get and print the details of each book
String title = element.getElementsByTagName("title").item(0).getTextContent();
String author =  element.getElementsByTagName("author").item(0).getTextContent();
String year =  element.getElementsByTagName("year").item(0).getTextContent();
String genre =  element.getElementsByTagName("genre").item(0).getTextContent();

System.out.println("Title: " + title);
 System.out.println("Author: " + author);
 System.out.println("Year: " + year);
 System.out.println("Genre: " + genre);
 System.out.println("-----------");
      }
   }
}
catch (Exception e) {
   e.printStackTrace();
      }
   }
}
```

Run the Program:

o Run the XmlParser.java class, and you should see the details of each book printed to the console.

```
Output
books.xml (run)   Java DB Database Process   GlassFish Server 4.1

run:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Year: 1925
Genre: Fiction
-----------
Title: To Kill a Mockingbird
Author: Harper Lee
Year: 1960
Genre: Fiction
-----------
Title: 1984
Author: George Orwell
Year: 1949
Genre: Dystopian
-----------
BUILD SUCCESSFUL (total time: 1 second)
```

Modifying XML Data

In this part, we will update the XML content programmatically using Java.

1. **Modify the XML Document**:

o In the XmlParser.java class, add code to update the **year** of the first book in the XML:

```
import java.io.File;

import java.io.InputStream;

import javax.xml.parsers.*;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import org.w3c.dom.*;


public class XmlParser {
```

```
public static void main(String[] args) {
    try {
        // Load XML from the src/xmlproject folder
        InputStream inputStream = XmlParser.class.getResourceAsStream("books.xml");

        if (inputStream == null) {
            System.out.println("File not found in package xmlproject!");
            return;
        }

        // Create a DocumentBuilderFactory and parse the XML content
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder builder = factory.newDocumentBuilder();
        Document document = builder.parse(inputStream);

        // Normalize document
        document.getDocumentElement().normalize();

        // Get all <book> elements
        NodeList nodeList = document.getElementsByTagName("book");

        // Loop through each book
        for (int i = 0; i < nodeList.getLength(); i++) {
            Node node = nodeList.item(i);

            if (node.getNodeType() == Node.ELEMENT_NODE) {
                Element element = (Element) node;

                // Extract values for each book
```

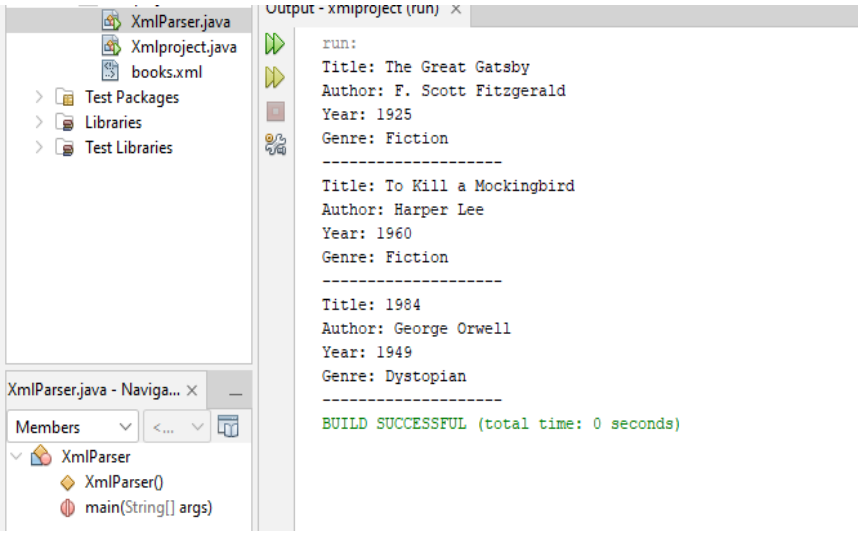
```
            String title = element.getElementsByTagName("title").item(0).getTextContent();
            String author = element.getElementsByTagName("author").item(0).getTextContent();
            String year = element.getElementsByTagName("year").item(0).getTextContent();
            String genre = element.getElementsByTagName("genre").item(0).getTextContent();

            // Print book details
            System.out.println("Title: " + title);
            System.out.println("Author: " + author);
            System.out.println("Year: " + year);
            System.out.println("Genre: " + genre);
            System.out.println("--------------------");
        }
    }
    Element firstBook = (Element) nodeList.item(0);
    firstBook.getElementsByTagName("year").item(0).setTextContent("2023");
    TransformerFactory transformerFactory = TransformerFactory.newInstance();
    Transformer transformer = transformerFactory.newTransformer();
    DOMSource source = new DOMSource(document);
    StreamResult result = new StreamResult(new File("updated_books.xml"));
    transformer.transform(source, result);




  } catch (Exception e) {
    e.printStackTrace();
  }
 }
}
```

output


XmlParser.java
Xmlproject.java
books.xml
Test Packages
Libraries
Test Libraries
XmlParser.java - Naviga...
Members
XmlParser
XmlParser()
main(String[] args)
Output - xmlproject (run)
run:
Title: The Great Gatsby
Author: F. Scott Fitzgerald
Year: 1925
Genre: Fiction

Title: To Kill a Mockingbird
Author: Harper Lee
Year: 1960
Genre: Fiction

Title: 1984
Author: George Orwell
Year: 1949
Genre: Dystopian

BUILD SUCCESSFUL (total time: 0 seconds)