Lab Sheet: Introduction to XML and Basic Operations

Lab Tasks:

Part 2: Creating Your First XML Document

1. Create an XML Document:

```
🔾 File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
                                                                                                      XMLExample - Apache NetBeans IDE 18
🔁 🛅 📮 : 🐚 📵 : | <default config> 🗸 📦 - 🚡 👸 🌓 - 🥳 🚮 - 🕦 - : | 12859/358.0MB 📞 📞
Projects ×
                          Start Page × 🚳 XMLExample.java × 🖫 book.xml ×
> 🦫 Area
> 🏐 dsa
                          <?xml version="1.0" encoding="UTF-8"?>
> b JavaApplication24
> SavaApplication41
                               Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
> 🦢 JavaApplication43
                               Click nbfs://nbhost/SystemFileSystem/Templates/XML/XMLDocument.xml to edit this template
JDBCExample
 inearSearch
 linearSearch1
 MultiThreadAp
  Pholymorphism
                          9 - <root>
                          10 E
                                    dibrary>
   Source Packages
                                    <book>
      mlexample xmlexample
                                   <title>The Great Gatsby</title>
<author>F. Scott Fitzgerald</author>
<year>1925</year>
<genre>Fiction</genre>
                          12
        XMLExample.java
                          13
        book.xml
                          14
 > 🍃 Libraries
                          15
                          16
                                    </book>
                          17
                          18
                                     <title>To Kill a Mockingbird</title>
                          19
                                     <author>Harper Lee</author>
                          20
                                     <year>1960</year>
                          21
                                    <genre>Fiction</genre>
XMLExample - Navigator ×
                          22
                                    </book>
                          23
                                    <book>
                          24
                                     <title>1984</title>
                          25
                                     <author>George Orwell</author>
                          26
                                     <year>1949</year>
                                    <genre>Dystopian</genre>
                          27
                          28
                                    </book>
                                  </library>
                          29
    <No View Available>
                                </root>
                          30
                          31
                          32
```

Part 3: Parsing XML in Java Create a Java Class for XML Parsing:

```
package xmlexample;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.io.File;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
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
      File xmlFile = new File("C:\\NAVEEN
GAYASHAN\\HNDIT\\XMLExample\\src\\xmlexample\\book.xml");
      Document document = builder.parse(xmlFile);
      // Normalize the document
      document.getDocumentElement().normalize();
      // Get the list of <book> elements
      NodeList nodeList = document.getElementsByTagName("book");
      // Loop through each <book> element
```

```
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 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();
    }
  }
}
```

output

Output - XMLExample (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

XML file updated and saved as updated_books.xml

BUILD SUCCESSFUL (total time: 0 seconds)

Part 4: Modifying XML Data Modify the XML Document:

```
package xmlexample;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.io.File;
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
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
      File xmlFile = new File("C:\\NAVEEN
GAYASHAN\\HNDIT\\XMLExample\\src\\xmlexample\\book.xml");
      Document document = builder.parse(xmlFile);
      // Normalize the document
      document.getDocumentElement().normalize();
      // Get the list of <book> elements
      NodeList nodeList = document.getElementsByTagName("book");
```

```
// Loop through each <book> element
      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 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("----");
        }
      }
        Element firstBook = (Element) nodeList.item(0);
      firstBook.getElementsByTagName("year").item(0).setTextContent("2023");
      // | Save the modified document to updated books.xml
      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);
      System.out.println("XML file updated and saved as updated books.xml");
```

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

// Save the modified document
}
```

output

Output - XMLExample (run)



run:



Title: The Great Gatsby
Author: F. Scott Fitzgerald



Year: 1925 Genre: Fiction

S _____

Title: To Kill a Mockingbird

Author: Harper Lee

Year: 1960 Genre: Fiction

Title: 1984

Author: George Orwell

Year: 1949

Genre: Dystopian

XML file updated and saved as updated_books.xml

BUILD SUCCESSFUL (total time: 0 seconds)

```
▼<library>
 ▼<book>
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald
    <year>1925</year>
    <genre>Fiction
   </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>
```