Lab Exercises

CSF205 – Data Representation, Markup Languages, and Web Services

Lab 05: XSLT and XML Programming

Resources to use:

Files required for the labs are provided in a separate directory.

XML Validator: https://www.xmlvalidation.com/

XML Beautifier: https://codebeautify.org/xmlviewer

XPath Browser: http://www.qutoric.com/xslt/analyser/xpathtool.html , http://xpather.com/

XSLT Test Tool: https://xslttest.appspot.com/

Tasks:

1. Open courses.xml and analyze the contents. Write the XSLT code for the following applying them on courses.xml. Starter template for XSLT is provided for each of them.

File: ex1.xsl **(Mandatory for Sign Off)** Return a list of department titles.
File: ex2.xsl **(Mandatory for Sign Off)** Return a list of department elements with no attributes and two sub-elements each: the department title and the entire Chair sub-element structure.
File: ex3.xsl Return all courses with enrollment greater than 500. Retain the structure of Course elements from the original data.

File: ex4.xsl
Remove from the data all courses with enrollment greater than 60, or with no enrollment listed. Otherwise the structure of the data should be the same.

File: ex5.xsl
Create a summarized version of the EE part of the course catalog. For each course in EE, return a Course element, with its Number and Title as attributes, its Description as a sub-element, and the last name of each instructor as an Instructor sub-element. Discard all information about department titles, chairs, enrollment, and prerequisites, as well as all courses in departments other than EE. (Note: To specify quotes within an already-quoted XPath expression, use quot;)

2. The files required for the XML Programming exercises given below are available in separate folders.
   2.1. Task 1 **(Mandatory for Sign Off)**

Folder: Task1 (consists of Height.java, cardlist.xml, recipes.xml)

Compile the file “Height.java”. Use the compiled program with various XML files as input to answer the following:

- What is the height of the XML tree for the recipes.xml file?

- What is the height of the XML tree for the cardlist.xml file?

Modify the “Height.java” source code so that it prints the number of “ingredient” elements that occur within the file that it is given.

Hint: the startElement method has a parameter called localName. This parameter will be set to “ingredient” if the current element being parsed is of type ingredient.

Lab Exercises

2.2. Task 2

Folder: Task2 (consists of BCeditjava, cardlist.xml and the jdom jar file to add to the classpath)

To compile and run the business card editing program (BCedit.java) with the “cardlist.xml” file you will need to include the JDOM class files from “jdom-2.0.5.jar” into the classpath. To compile the program you will need to use:

```
javac -cp jdom-2.0.5.jar BCedit.java
```

Note: The compilation process may give you several warnings, please ignore them.

To run the program you will need to use:

```
java -cp jdom-2.0.5.jar;. BCedit cardlist.xml
```

Remember to use jdom-2.0.5.jar**:.** (which has a colon (:) instead of semicolon (;)) in Linux or Mac machines. Semicolon (;) is used only on Windows.

- In the BCedit.java you will find some //TODO statements which you need to fix in order for the code to compile
- Once it compiles successfully, run the java code, click on “new” button and enter new values for the business card and press “ok”. This will list the newly added business card to the list on the left hand side. If it is successful, press “save” to save it to the cardlist.xml. Press “quit” and reopen the application and check if you are able to see the newly added business card in the list.