CO3098/CO7098
Coursework 1

XML Schema, XSLT and XML Parsing

Important Dates:

Handed out: 7-Oct-2016
Deadline: 28-Oct-2016 at 23:59 GMT

The deadline is strict and will not be changed. Please ensure that you submit your work in time.

- This coursework counts as 8% of your final mark.
- Please read guidelines on plagiarism in the study guide and course documentation.
- This coursework requires knowledge about XML Schema, XSLT and DOM/SAX Parsing
- Please put your plagiarism coversheet for this coursework in Box B or submit a signed coversheet electronically to the blackboard.

Data Decription

Given the following XML document (WebService.xml):

```
<?xml version="1.0" encoding="UTF-8"?>
<interface name="WebServiceInterface">
          <package>uk.ac.le.cs.wt</package>
          <import>java.io.*</import>
          <import> java.rmi.RemoteException</import>
          <abstract_method name="login">
             <modifier>public</modifier>
             <arguments>
                <parameter type="int">userid</parameter>
                <parameter type="String">password</parameter>
             </arguments>
             <return>boolean</return>
          </abstract_method>
          <abstract_method name="sendEmail">
             <modifier>public</modifier>
             <arguments>
                <parameter type="String">recipient</parameter>
                <parameter type="String">message</parameter>
             </arguments>
             <exceptions>
                <exception>IOException</exception>
                <exception>RemoteException</exception>
             </exceptions>
             <return>void</return>
          </abstract_method>
</interface>
```

This XML document provides an alternative representation of a **Java interface** below

```
package uk.ac.le.cs.wt;

import java.io.*;
import java.rmi.RemoteException;

interface WebServiceInterface{

   public boolean login(int userid, String password);
   public void sendEmail(String recipient, String message)
               throws IOException, RemoteException ;

}
```

Tasks:

Task 1: [35 Marks]

Write an XML Schema (WebService.xsd) that allows validation of the provided XML document. Note that the schema must be consistent with these rules in Java:

- There can be **only one** package statement
- There can be **zero to many** import statements
- An interface can have **zero to many** abstract methods
- A method only allows **one** value to be returned
- A method can only have **one** visibility modifier
- A methods can accept **zero to many** parameters
- Exception(s) may be thrown from a method

Task 2: [30 Marks]

Write an XSLT stylesheet (WebService.xslt) that takes the provided XML document as input and produces an HTML document as follows:

WebServiceInterface

Method	Return type	Input parameters
login	boolean	userid: int, password: String
sendEmail	void	recipient: String, message: String

(**Note**: You may show the result in a different page layout or colour scheme. The page must display the name, return type and the parameter list of each method)

Task 3: [35 Marks]

Write a Java program that reads the XML above and produces an output as follows. Please choose an appropriate parsing technology (DOM/SAX).

```
public boolean login(int userid, String password);

public void sendEmail(String recipient, String message) throws IOException,
RemoteException ;
```

(**Note**: You need to implement either WebServiceDOM.java or WebServiceSAX.java)

Submission

- Zip all files in a single zip file for submission.
    - WebService.xsd
    - WebService.xslt
    - WebServiceSAX.java (or WebServiceDOM.java, depending on your choice)
- The archive should be named **CO3098_CW1_email_id.zip** or **CO7098_CW1_email_id.zip** (e.g. CO3098_CW1_yh37.zip).

Your submission should also include a completed coursework plagiarism coversheet (print and signed PDF or image). You need to submit the zip file via Blackboard and you are allowed to re-submit as many times as you like **before** the deadline. Marks for any coursework which does not have the accompanying cover sheet will be withheld till you provide one.

Note: apart from the XML document above, we might use other XML instance documents to evaluate Task 1, 2 and 3. The XML document should also be validated against your XSD schema, and the XPATH expressions used in your XSLT should also work for the XML document we provided.