

# COMPSCI 732 FC 2005

## Homework: XSL Transforms

19 May 2005

For this homework you will look at creating a XSL transform to visualise a XML document containing course information for a department (as developed in Tuesday's homework) in HTML. The XML document containing course information, and its corresponding DTD, can be found on this course's web site for this lab. You must create a XSL transform which provides HTML similar to that shown on the other side of this handout.

Note the processing instruction in the XML document:

```
<?xml:stylesheet type="text/xsl" href="DepartmentSchedule.xsl"?>
```

This requires that the XSL be placed in a file called `DepartmentSchedule.xsl`

Your XSL should create a HTML file with a structure that looks something like the following:

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1></h1>
    <h2></h2>
    <table>
      <tr><td>...</td><td>...</td><td>...</td></tr>
      ...
    </table>
  </body>
</html>
```

Note that the HTML tags above have the following properties:

- `title`: provides the string for the browser window's title
- `h1`: provides a top level heading
- `h2`: provides a secondary level heading
- `table`: creates a table
  - `tr`: denotes the start of a row in a table
  - `td`: denotes the start of a column within a row
- `b`: makes an element bold text

Note that you can visualise your XSL transform by viewing the XML document within Internet Explorer (IE). IE will use the specified `.xsl` file to render this XML document.

Also note that a XSL file is specified in XML format so needs a header something like:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

Course Schedule - Software Engineering - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

# Software Engineering (2005)

## Course Schedule

Course Code	Name	Description
SOFTENG 206 DC	Engineering Design 2	<p>This course provides experience for project work. Students work on projects over two semesters, and will be exposed to a number of topics relating to the systematic production of software.</p> <p>This paper looks in the first half of the course at extending the programming-in-the-small concepts and practices from SOFTENG 250. The Unified Modelling Language is introduced as a program design notation, and object-oriented programming concepts like inheritance are introduced. Various supporting ideas like coding standards, refactoring, unit testing, debugging and program documentation are learned. An introduction to design patterns is given. In the second half of the course programming-in-the-medium concepts and skills are developed. These include the use of more UML constructs and processes for life cycle phases of requirements, specification and design, and integration and system testing. Use databases and basic Java web facilities to build distributed, on-line software systems. System documentation, version control and peer review are also introduced.</p>
SOFTENG 251 SC	Software Engineering 1	<p>To provide the student with an understanding of advanced algorithm design and the analytical and empirical behaviour of advanced algorithms and data structures.</p>
SOFTENG 253 SC	Algorithms	

Done My Computer