Exercise sheet 2
Recuperación de Información: XPath & XSLT

1) Taking as input the **students.xml** file (attached in the zip file), provide XPath expressions for the following statements.

a) Select all the elements in the document
b) Select all students
c) Select the name last name second student of the document
d) Select the name of the student with id "2028"
e) Select the last student of the document
f) Select the name, last name all the students
g) Select all the students ids
h) Select the name, last name and telephone of the male student
i) Select the name and last name of the students that are males or study the master degree
j) Select all the students who have a telephone number

Last name	*First name*	*ID*	*Master*	*Study*	*Gender*	*email*	*Telephone*
Sielmann	Gustav	2028	Yes	Architecture	Male	gsielmann@university.org	6045
Rummer	Arnold	2037	Yes	Architecture	Male	arummer@ university.org	6085
Neumeier	Johanna	5042	No	Informatics	Female	jneumeier@ university.org	6093
Müller	Patricia	5048	Yes	Architecture	Female	pmueller@ university.org	
Thaler	Karl	7123	No	Business	Male	kthaler@ university.org	6152

2) XPath: Given the following XML Document:

```
<?xml version="1.0" encoding="UTF-8"?>
<lehre>
  <responsible>
    <title>Dr.</title>
    <name>Ying Ding</name>
  </responsible>
  <course sine-tempore="no" year="2003" type="lecture">
    <title>Telecooperation</title>
    <lecturer>Dieter Fensel</lecturer>
    <exam>
      <date>2.7.</date>
      <room>HS A</room>
      <max>150</max>
    </exam>
  </course>
  <course sine-tempore="yes" year="2004" type="projectLab">
    <title>Semantic Web</title>
    <lecturer>Ying Ding</lecturer>
    <lecturer>Axel Polleres</lecturer>
  </course>
  <course sine-tempore="yes" year="2005" type="projectLab">
    <title>Semantic Web</title>
    <lecturer>Thomas Strang</lecturer>
    <lecturer>Axel Polleres</lecturer>
  </course>
  <course sine-tempore="no" year="2005" type="lecture">
    <title>Telecooperation</title>
    <lecturer>Axel Polleres</lecturer>
    <exam>
      <date>1.7.</date>
      <room>HS A</room>
      <max>150</max>
    </exam>
    <exam>
      <date>29.9.</date>
      <room>HS B</room>
      <max>150</max>
    </exam>
  </course>
  <course sine-tempore="yes" year="2005" type="tutorial">
    <title>Telecooperation UE</title>
    <lecturer>Sinuhe Arroyo</lecturer>
    <lecturer>Ioan Toma</lecturer>
    <lecturer>Holger Lausen</lecturer>
    <lecturer>Anna Zhdanova</lecturer>
   </course>
</lehre>
```

Write down XPath expressions for the following queries:

a) all courses of type "projectlab" where Thomas Strang was not a lecturer
b) the first course which has more than one exam
c) all titles of lectures before 2005

3) Given the following XML and XSLT, write the resulting document:

```
<?xml version="1.0"?>
<businessCards>
   <card>
        <name>John Doe</name>
        <title>CEO, Widget Inc.</title>
        <email>john.doe@widget.com</email>
   </card>

   <card type="simple">
        <name>Max Muster</name>
        <title>Management Director, Widget Inc.</title>
        <email>max.muster@widget.com</email>
   </card>
</businessCards>
```

XSLT:

```
 <?xml version="1.0"?>
      <xsl:stylesheet xmlns:xsl=http://www.w3.org/1999/XSL/Transform
                      version="1.0"
xmlns="http://www.w3.org/1999/xhtml">

  <xsl:template match="/">
      <html xmlns="http://www.w3.org/1999/xhtml">
      <xsl:apply-templates/>
      </html>
  </xsl:template>

  <xsl:template match="card[@type='simple']">
      <xsl:apply-templates select="title"/>
      <body>
      <xsl:apply-templates select="name"/>
      <xsl:apply-templates select="email"/>
      </body>
   </xsl:template>

   <xsl:template match="card[not(@type='simple')]"/>

   <xsl:template match="name">
      <xsl:element name="h2">
      <xsl:attribute name="id">123</xsl:attribute>
      <xsl:value-of select="text()"/>
      </xsl:element>
   </xsl:template>

   <xsl:template match="email">
      <p>email: <a href="mailto:{text()}"><tt>
      <xsl:value-of select="text()"/>
      </tt></a></p>
   </xsl:template>

   <xsl:template match="title">
      <xsl:element name="title">
      <xsl:value-of select="text()"/>
      </xsl:element>
   </xsl:template>
</xsl:stylesheet>
```

Result:

4) XSLT: Given the following XSLT stylesheet and XML source file, complete the resulting XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="xml" version="1.0" encoding="UTF-
8“ indent=“yes”/>

<xsl:template match="/">
  <xyz>
    <xsl:apply-templates/>
  </xyz>
</xsl:template>

<xsl:template match="test1">
    <test>
       <att1>
          <xsl:value-of select="@att1"/>
       </att1>
       <xsl:apply-templates/>
    </test>
</xsl:template>

<xsl:template match="test2">
   <xsl:element name="test123">
     <xsl:attribute name="att2">
       <xsl:value-of select="."/>
     </xsl:attribute>
   </xsl:element>
</xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<document>
  <test1 att1="Hallo">
      <test2>Welt</test2>
  </test1>
  <test1 att0="Guten" att1="Tag" att2="miteinander">
    <test2>Globus</test2>
  </test1>
</document>
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

5) Taking as input the **students.xml** file (attached in the zip file), provide XSLT style sheet that generates an HTML document that lists the information in tabular form. You can test your transformation using XMLSpy or any other tool of your choice

6) Freestyle: Find an appropriate page to wrap and extract an XML file.
For instance, you could try to write a generic XSLT file which extracts the temperature and Wind direction from

http://weather.yahoo.com/

for different cities. E.g.
Madrid is http://weather.yahoo.com/forecast/SPXX0050.html
Mostoles is http://weather.yahoo.com/forecast/SPXX0050.html
Galway is http://weather.yahoo.com/forecast/EIXX0017.html

For this exercise, you can use tools like

W3C Tidy:
http://tidy.sourceforge.net/
http://jtidy.sourceforge.net/
XSLT processor e.g. Apache XALAN:
Java: http://xml.apache.org/xalan-j/index.html
C++: http://xml.apache.org/xalan-c/index.html

If you do 6), you don't have to do 1) and 5) ! ;-)