1. Do the workshop in the chapter 7 & 8
2. After finishing the workshop students must complete the following tasks:
   More on XSLT
   - XPATH and Expressions functions

   DOM
   - Loading XML document
   - Accessing nodes
   - Attribute list

3. More on XSLT

a. XPATH and Expressions functions

Type and save the following codes as in turn **CheckBook.xml** and **CheckBook.xsl** files:

```
<?xml version="1.0"?>

<?xml-stylesheet type="text/xsl" href="checkbook.xsl" ?>
<checkbook>

  <deposit type="direct-deposit">
    <payor>Bob's Bolts</payor>
    <amount>987.32</amount>
    <date>21-6-00</date>
    <description category="income">Paycheck</description>
  </deposit>

  <payment type="check" number="980">
    <payee>Kimora's Sports Equipment</payee>
    <amount>132.77</amount>
    <date>23-6-00</date>
    <description category="entertainment">kendo equipment</description>
  </payment>

  <payment type="atm">
    <amount>40.00</amount>
    <date>24-6-00</date>
    <description category="cash">pocket money</description>
  </payment>

  <payment type="debit">
    <payee>Lone Star Cafe</payee>
    <amount>36.86</amount>
    <date>26-6-00</date>
    <description category="food">lunch with Greg</description>
  </payment>

  <payment type="check" number="981">
    <payee>Wild Oats Market</payee>
    <amount>47.28</amount>
    <date>29-6-00</date>
    <description category="food">groceries</description>
  </payment>

  <payment type="debit">
    <payee>Barnes and Noble</payee>
```

```
    <amount>58.79</amount>
    <date>30-6-00</date>
    <description category="work">O'Reilly Books</description>
  </payment>

  <payment type="check" number="982">
    <payee>Old Man Ferguson</payee>
    <amount>800.00</amount>
    <date>31-6-00</date>
    <description category="misc">a 3-legged antique credenza that once
    belonged to Alfred Hitchcock</description>
  </payment>

</checkbook>
```

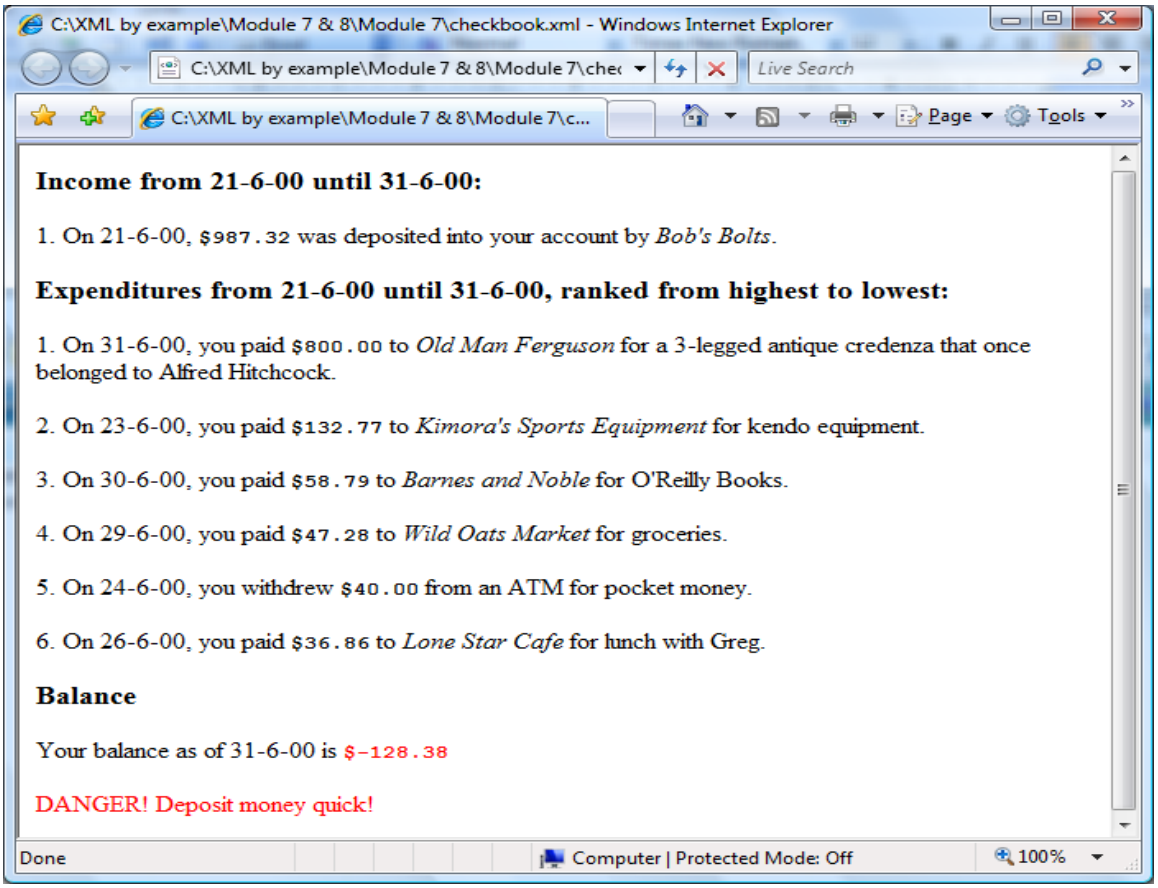
```
<?xml version="1.0"?>

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

<xsl:template match="checkbook">
  <html>
    <head/>
    <body>
      <h3>
        <xsl:text>Income from </xsl:text>
        <xsl:value-of select="child::*[1]/date"/>
        <xsl:text> until </xsl:text>
        <xsl:value-of select="child::*[last()]/date"/>
        <xsl:text>:</xsl:text>
      </h3>
      <xsl:apply-templates select="deposit"/>
      <h3>
        <xsl:text>Expenditures from </xsl:text>
        <xsl:value-of select="child::*[1]/date"/>
        <xsl:text> until </xsl:text>
        <xsl:value-of select="child::*[last()]/date"/>
        <xsl:text>, ranked from highest to lowest:</xsl:text>
      </h3>
      <xsl:apply-templates select="payment">
        <xsl:sort data-type="number" order="descending" select="amount"/>
      </xsl:apply-templates>
      <h3>Balance</h3>
      <p>
        <xsl:text>Your balance as of </xsl:text>
        <xsl:value-of select="child::*[last()]/date"/>
        <xsl:text> is </xsl:text>
        <tt><b>
          <xsl:choose>
            <xsl:when test="sum( payment/amount )> sum( deposit/amount
)">
              <font color="red">
                <xsl:text>$</xsl:text>
                <xsl:value-of select="sum( deposit/amount )
```

```
                                        - sum( payment/amount )"/>
              </font>
            </xsl:when>
            <xsl:otherwise>
              <font color="blue">
                <xsl:text>$</xsl:text>
                <xsl:value-of select="sum( deposit/amount )
                                        - sum( payment/amount )"/>
              </font>
            </xsl:otherwise>
          </xsl:choose>
        </b></tt>
      </p>
      <xsl:if test="sum( payment/amount )> sum( deposit/amount )">
        <p>
          <font color="red">
            <xsl:text>DANGER! Deposit money quick!</xsl:text>
          </font>
        </p>
      </xsl:if>
    </body>
  </html>
</xsl:template>

<xsl:template match="payment[@type='atm']">
  <p>
    <xsl:value-of select="position()"/>
    <xsl:text>. On </xsl:text>
    <xsl:value-of select="date"/>
    <xsl:text>, you withdrew </xsl:text>
    <tt><b>
      <xsl:text>$</xsl:text>
      <xsl:value-of select="amount"/>
    </b></tt>
    <xsl:text> from an ATM for </xsl:text>
    <xsl:value-of select="description"/>
    <xsl:text>.</xsl:text>
  </p>
</xsl:template>

<xsl:template match="payment">
  <p>
    <xsl:value-of select="position()"/>
    <xsl:text>. On </xsl:text>
    <xsl:value-of select="date"/>
    <xsl:text>, you paid </xsl:text>
    <tt><b>
      <xsl:text>$</xsl:text>
      <xsl:value-of select="amount"/>
    </b></tt>
    <xsl:text> to </xsl:text>
    <i>
      <xsl:value-of select="payee"/>
    </i>
    <xsl:text> for </xsl:text>
```

```
    <xsl:value-of select="description"/>
    <xsl:text>.</xsl:text>
  </p>
</xsl:template>

<xsl:template match="deposit">
  <p>
    <xsl:value-of select="position()"/>
    <xsl:text>. On </xsl:text>
    <xsl:value-of select="date"/>
    <xsl:text>, </xsl:text>
    <tt><b>
      <xsl:text>$</xsl:text>
      <xsl:value-of select="amount"/>
    </b></tt>
    <xsl:text> was deposited into your account by </xsl:text>
    <i>
      <xsl:value-of select="payor"/>
    </i>
    <xsl:text>.</xsl:text>
  </p>
</xsl:template>
</xsl:stylesheet>
```

Open **CheckBook.xml** file in the browser

The output:



4. XML DOM: (*note : using the Internet Explorer browser*)

a. Loading XML DOM

Type and save the following codes as in turn **Book.xml** and **LoadingXMLDom.html** files:

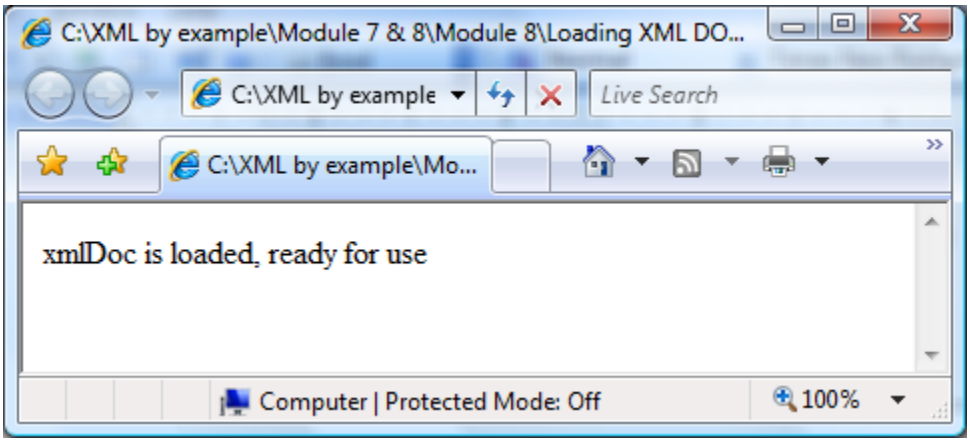
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<bookstore>
<book category="cooking">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="children">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="web">
```

```
<title lang="en">XQuery Kick Start</title>
<author>James McGovern</author>
<author>Per Bothner</author>
<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<year>2003</year>
<price>49.99</price>
</book>
<book category="web" cover="paperback">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

```
<html>
<body>
<script type="text/javascript">
try //Internet Explorer
  {
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  }
catch(e)
  {
  try //Firefox, Mozilla, Opera, etc.
    {
    xmlDoc=document.implementation.createDocument("","",null);
    }
  catch(e) {alert(e.message)}
  }
try
  {
  xmlDoc.async=false;
  xmlDoc.load("books.xml");
  document.write("xmlDoc is loaded, ready for use");
  }
catch(e) {alert(e.message)}
</script>
</body>
</html>
```

Open **LoadingXMLDom.xml** file in the browser

The output:



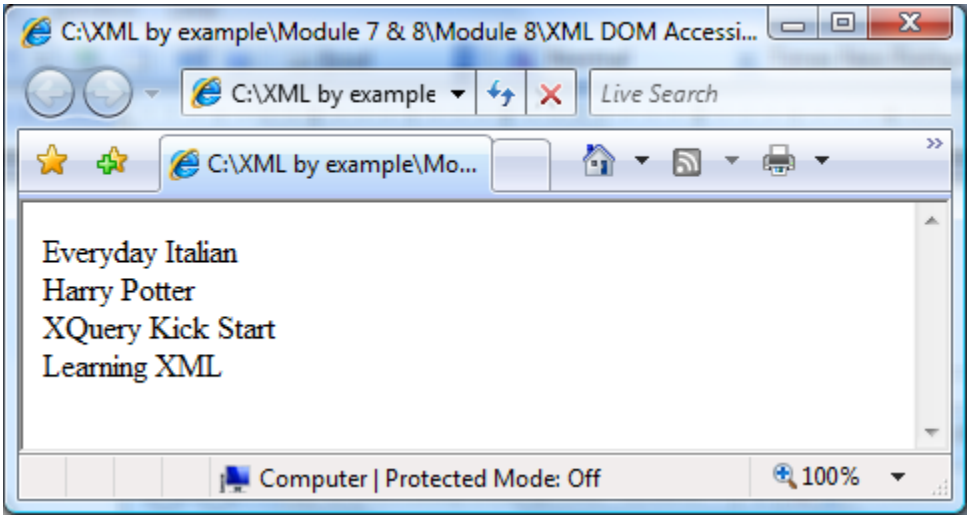
b. Accessing Nodes

This example using the **Book.xml** file showed above. Type and save the following code as **AccessingNodes.html** file and put this file in the same directory with the **Book.xml**:

```
  <html>
<head>
<script>
function loadXMLDoc(dname)
{
try //Internet Explorer
  {
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  }
catch(e)
  {
  try //Firefox, Mozilla, Opera, etc.
    {
    xmlDoc=document.implementation.createDocument("","",null);
    }
  catch(e) {alert(e.message)}
  }
try
  {
  xmlDoc.async=false;
  xmlDoc.load(dname);
  return(xmlDoc);
  }
catch(e) {alert(e.message)}
return(null);
}
</script>
</head>
<body>

<script type="text/javascript">
xmlDoc=loadXMLDoc("books.xml");
```

```
x=xmlDoc.getElementsByTagName("title");
for (i=0;i<x.length;i++)
  {
  document.write(x[i].childNodes[0].nodeValue);
  document.write("<br />");
  }
</script>
</body>
</html>
```

Run this file in the Internet Explorer browser.
The output:



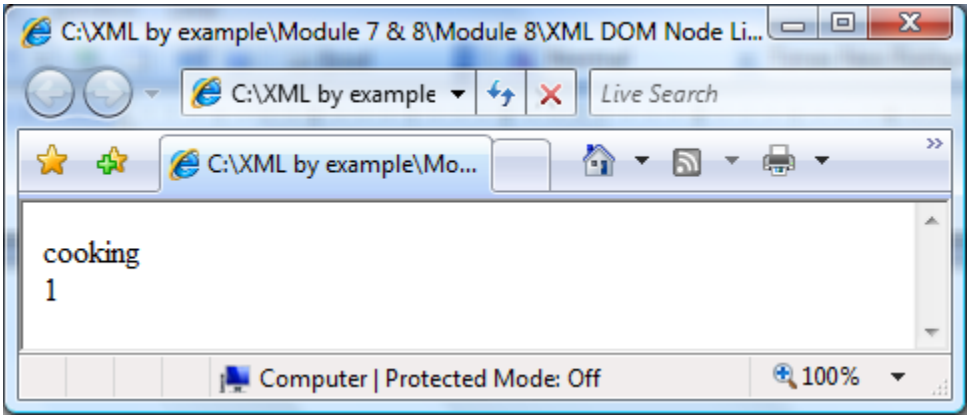
c. Attribute list

This example using the **Book.xml** file showed above. Type and save the following code as **AttributeList.html** file and put this file in the same directory with the **Book.xml**:

```
<html>
<head>
<script>
function loadXMLDoc(dname)
{
try //Internet Explorer
  {
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
  }
catch(e)
  {
  try //Firefox, Mozilla, Opera, etc.
    {
    xmlDoc=document.implementation.createDocument("","",null);
    }
  catch(e) {alert(e.message)}
```

```
  }
try
  {
  xmlDoc.async=false;
  xmlDoc.load(dname);
  return(xmlDoc);
  }
catch(e) {alert(e.message)}
return(null);
}
</script>
</head>
<body>

<script type="text/javascript">
xmlDoc=loadXMLDoc("books.xml");

x=xmlDoc.getElementsByTagName("book")[0].attributes;
document.write(x.getNamedItem("category").nodeValue);
document.write("<br />" + x.length);
</script>
</body>
</html>
```

Run this file in the Internet Explorer browser.
The output:



Do It Yourself

4.1. Using the XML DOM access and display the Sales amount of the last <SalesPerson> node in the following xml file:

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

<?xml-stylesheet type="text/xsl" href="SalesData_stylesheet.xsl"?>
```

```
<CottonDesk>

  <SalesPerson>

    <Product>Monitor</Product>

    <Name>Harry Diana</Name>

    <Age>23</Age>

    <Sales>4000</Sales>

  </SalesPerson>
  <SalesPerson>

    <Product>Laptop</Product>

    <Name>David Blake</Name>

    <Age>32</Age>

    <Sales>20000</Sales>

  </SalesPerson>

    <SalesPerson>

    <Product>Laptop</Product>

    <Name>Susan Jones</Name>

    <Age>55</Age>

    <Sales>35000</Sales>

  </SalesPerson>

    <SalesPerson>

    <Product>Laptop</Product>

    <Name>Martin Howell</Name>

    <Age>25</Age>

    <Sales>1000</Sales>

  </SalesPerson>
```

```
    <SalesPerson>
    <Product>Keyboard</Product>
    <Name>John Dani</Name>
    <Age>45</Age>
    <Sales>35000</Sales>
  </SalesPerson>
    <SalesPerson>
    <Product>Laptop</Product>
    <Name>Tony Kemp</Name>
    <Age>35</Age>
    <Sales>31000</Sales>
  </SalesPerson>
  </CottonDesk>
```

4.2. Navigate and count how many <product> nodes in the xml file.