

CSC343 Worksheet 11 Solution

June 30, 2020

```
1. a) 1 <? xml version = "1.0" encoding = "utf-8"?>
      2 <xsl:stylesheet xmlns:xsl = "http://www.w3.org/1999/ISL/Transform
      3 ">
      4   <xsl:template match = "/">
      5     <html>
      6       <BODY>
      7         <H1>Manufacturerers</h1>
      8         <OL>
      9           <xsl:for-each select="Products/Maker">
          10             <Li><xsl:value-of select="@name"/></li>
          11           </xsl:for-each>
          12         </ol>
          13       </body>
          14     </html>
          15   </xsl:template>
          16 </xsl:stylesheet>
```

Notes:

- XLST
 - Also means **Extensible Stylesheet Language for Transformation**
 - Allows to transform XML documents into HTML
 - Is another language
- XLST Basics

Syntax:

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

- Templates

Example:

```

1  <? xml version = "1.0" encoding = "utf-8"?>
2  <xsl:stylesheet xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
3      <xsl:template match = "/">
4          <HTML>
5              <BODY>
6                  <B>This is text</b>
7              </body>
8          </html>
9      </xsl:template>
10 </xsl:stylesheet>
11

```

- Obtaining values from XML Data

- value-of

- * Is used to extract the value of a selected node.

- * **Syntax:** <xsl value-of select = "*expression*" />

Example:

```

1  <? xml version = "1.0" encoding = "utf-8"?>
2  <xsl:stylesheet xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
3      <xsl:template match="/Movies/Movie">
4          <xsl:value-of select = "@title"/>
5      </xsl:template>
6  </xsl:stylesheet>
7

```

- Apply-templates

- * applies a <xsl:template> to the current element or to the current element's child nodes.

- select='.' means choose the current value

```

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

<xsl:template match="/*">
<html>
<body>
<h2>My CD Collection</h2>
<xsl:apply-templates/>
</body>
</html>
</xsl:template>

<xsl:template match="cd">
<p>
<xsl:apply-templates select="title"/>
<xsl:apply-templates select="artist"/>
</p>
</xsl:template>

<xsl:template match="title">
Title: <span style="color:#ff0000">
<xsl:value-of select="."/;></span>
<br />
</xsl:template>

<xsl:template match="artist">
Artist: <span style="color:#00ff00">
<xsl:value-of select="."/;></span>
<br />
</xsl:template>

</xsl:stylesheet>

```

My CD Collection

```

Title: Empire Burlesque
Artist: Bob Dylan

Title: Hide your heart
Artist: Bonnie Tyler

Title: Greatest Hits
Artist: Dolly Parton

Title: Still got the blues
Artist: Gary Moore

Title: Eros
Artist: Eros Ramazzotti

Title: One night only
Artist: Bee Gees

Title: Sylvias Mother
Artist: Dr.Hook

Title: Maggie May
Artist: Rod Stewart

Title: Romanza
Artist: Andrea Bocelli

Title: When a man loves a woman
Artist: Percy Sledge

Title: Black angel
Artist: Savage Rose

Title: 1999 Grammy Nominees
Artist: Many

Title: For the good times
Artist: Kenny Rogers

```

- For-loop in XSLT


- * **Syntax:** <xsl:for-each select = "*expression*" > ... </xsl:for-each>

* Is easier to maintain and read than apply templates ^[1]

```

1)  <? xml version = "1.0" encoding = "utf-8" ?>
2)  <xsl:stylesheet xmlns:xsl =
3)      "http://www.w3.org/1999/XSL/Transform">
4)      <xsl:template match = "/">
5)          <OL>
6)              <xsl:for-each select = "Stars/Star" />
7)                  <LI>
8)                      <xsl:value-of select = "Name">
9)                  </li>
10)             </xsl:for-each>
11)             </ol><p><ol>
12)             <xsl:for-each select = "Stars/Star/Address" />
13)                 <LI>
14)                     <xsl:value-of select = "City">
15)                 </li>
16)             </xsl:for-each>
17)             </ol>
18)         </xsl:template>
19)     </xsl:stylesheet>

```



References:

1) StackOverflow, differences between for-each and templates in xsl?, link

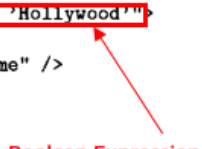
- If condition in XSLT

– **Syntax:** <xsl:if test = "*boolean expression*">

```

1)  <? xml version = "1.0" encoding = "utf-8" ?>
2)  <xsl:stylesheet xmlns:xsl =
3)      "http://www.w3.org/1999/XSL/Transform">
4)      <xsl:template match = "/">
5)          <TABLE border = "5"><TR><TH>Stars</th></tr>
6)          <xsl:for-each select = "Stars/Star" />
7)              <xsl:if test = "Address/City = 'Hollywood'">
8)                  <TR><TD>
9)                      <xsl:value-of select = "Name" />
10)                 </td></tr>
11)             </xsl:if>
12)         </xsl:for-each>
13)     </table>
14) </xsl:template>
15) </xsl:stylesheet>

```



b)

```

1  <? xml version = "1.0" encoding = "utf-8"?>
2  <xsl:stylesheet xmlns:xsl = "http://www.w3.org/1999/XSL/Transform"
3  >
4      <xsl:template match = "/">
5          <html>
6              <BODY>
7                  <TABLE>
8                      </table>
9              </body>
10             </html>
11         </xsl:template>
12     </xsl:stylesheet>
13

```