WHERE CAN YOU FIND YOUR FAVORATE CAFE?

(Maximum Points: 50)

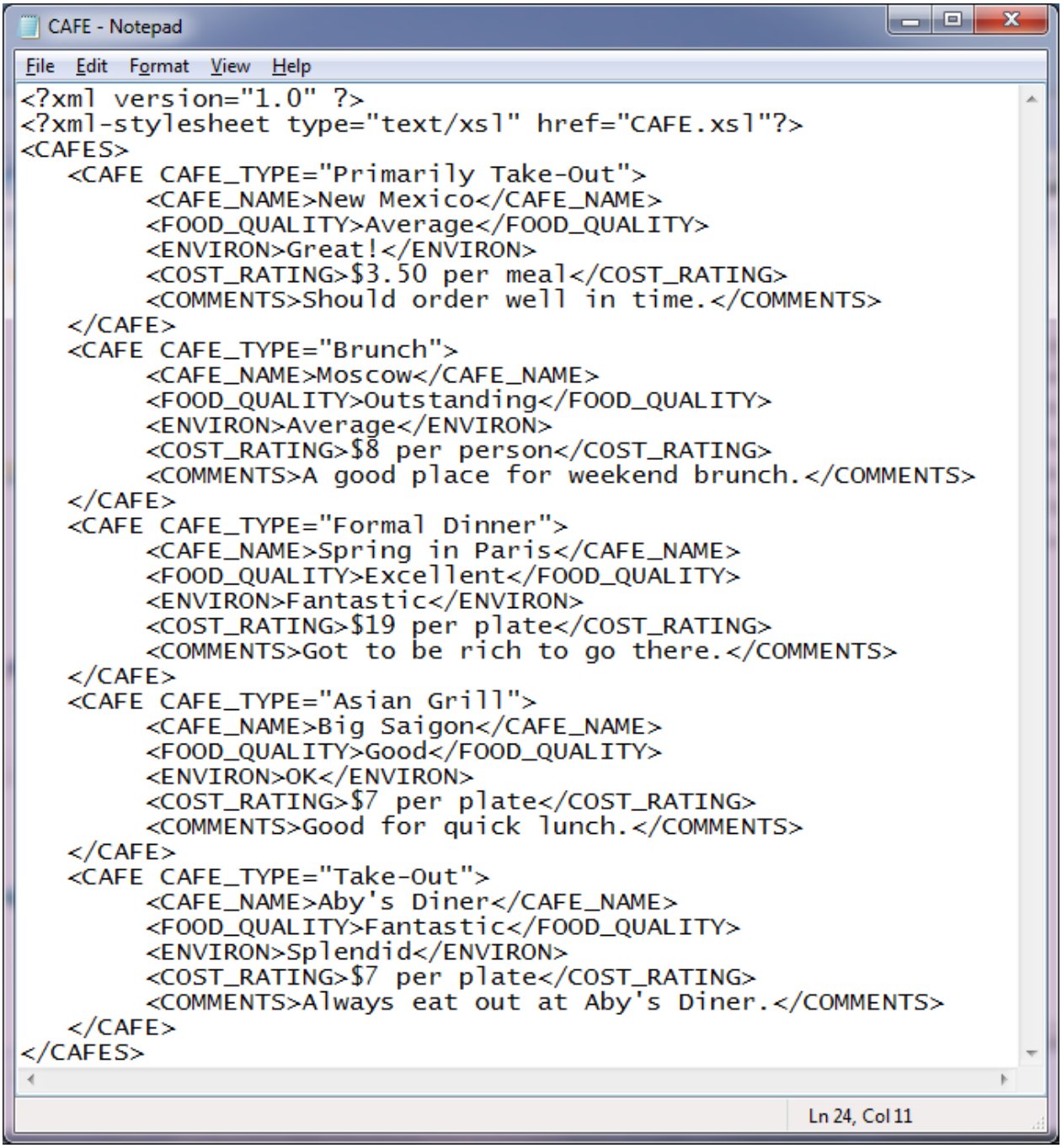
While HTML is a markup language with a fixed set of tags that allows users to specify the appearance of a document, XML (extensible markup language) allows the user to create new tags to provide a document structure appropriate to the task at hand. A document has three aspects: structure, appearance, and content. In the case of XML, these aspects are separated, with different means employed for defining the structure, the appearance, and the content. Usually, the document type definition (DTD) file describes the structure of a document; the appearance is specified by an extensible style sheet (XSL); and the content is provided in the XML document.

In the case of XML, the Web server does not send an HTML document but rather sends the XML and the XSL documents, and the client must have XML/XSL processors to display the document. In this assignment, you will learn to create a simple XML document first. You will then learn to specify the structure of the XML file and format your XML document using XSL.

1. Creating a Simple XML Document (20 points)

As a programmer, you are asked to provide a cafe comparison table for web surfers. In order to do that, you first need to create all the related cafe information using a simple XML document.

1. Open Notepad or any XML Editor on your computer system, create a file and name it CAFE.xml.
2. First define the XML version by writing “<?xml version="1.0" ?> ” in the first line.
3. The next line, “<?xml-stylesheet type="text/xsl" href="CAFE.xsl"?>”, refers to the type of style sheet used and the name of the file.
4. The structure of the XML file is simple. At the root is a structure called CAFES. This structure at the next level has a repeated number of child structures called CAFE. In turn, the structure CAFE has one attribute, CAFE_TYPE, and it has five elements:
    - CAFE_NAME
    - FOOD_QUALITY
    - ENVIRON
    - COST_RATING
    - COMMENTS
5. The code for the entire file CAFE.xml is in the following image:

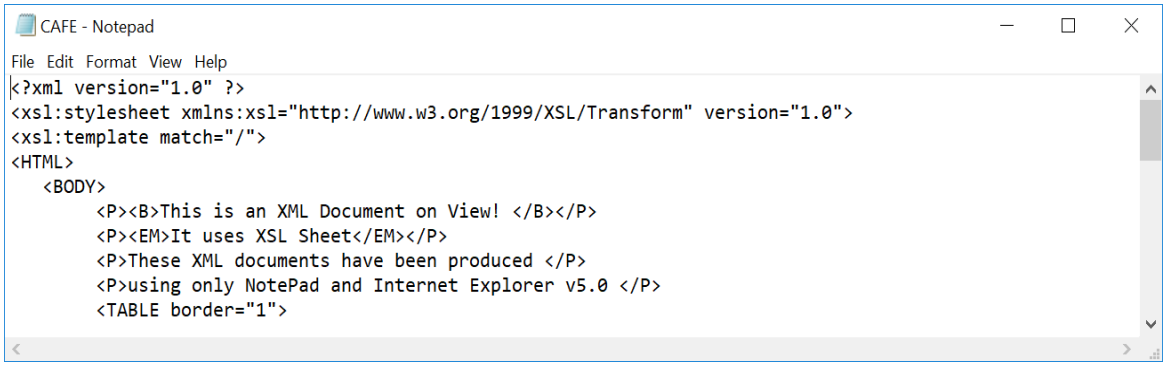


2. Formatting XML Using XSL (20 points)

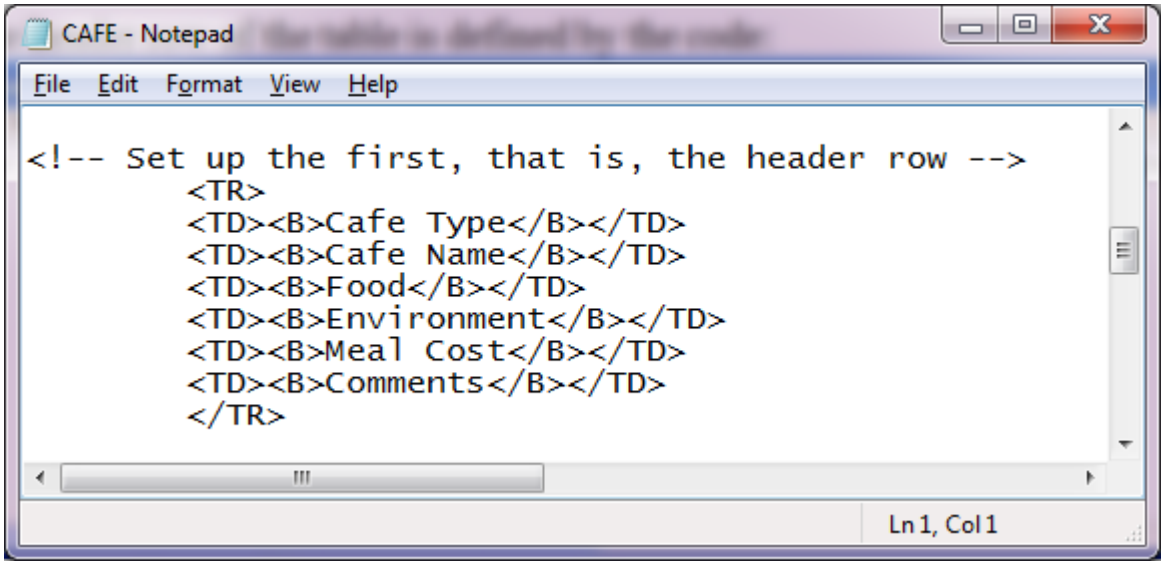
6. Now consider the XSL file, CAFE.xsl. The file accomplishes several tasks: (1) the XSL file generates the HTML code that eventually runs on the browser. The XSL file has to create the necessary HTML tags for that purpose. (2) As part of the above, the XSL code creates a table that has the tabular headings in the first row. (3) Looping constructs allowed in XSL are used to read individual elements of CAFE in the associated XML file and then to output the values in the HTML table.

7. Preamble. The file begins with a preamble that defines the location of the XSL specification by the following line: <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">. And, remember to add <?xml version="1.0" ?> as the first line. The URL as shown in the image below does not work any more.

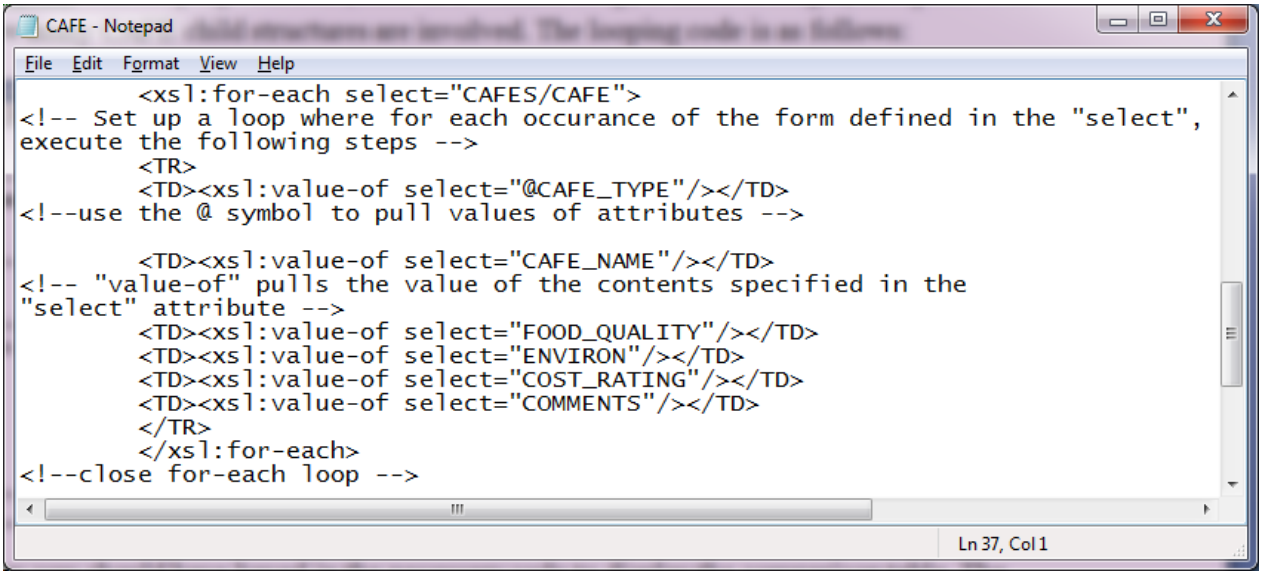
8. Template Match. The line: <xsl:template match="/"> instructs the system to match the XSL template right from the root stage of the structure described in the XML file (i.e., the entire structure of CAFES/CAFE is to be processed as per the XSL file).
9. The HTML tags for the file and for defining the table are achieved by the following code:



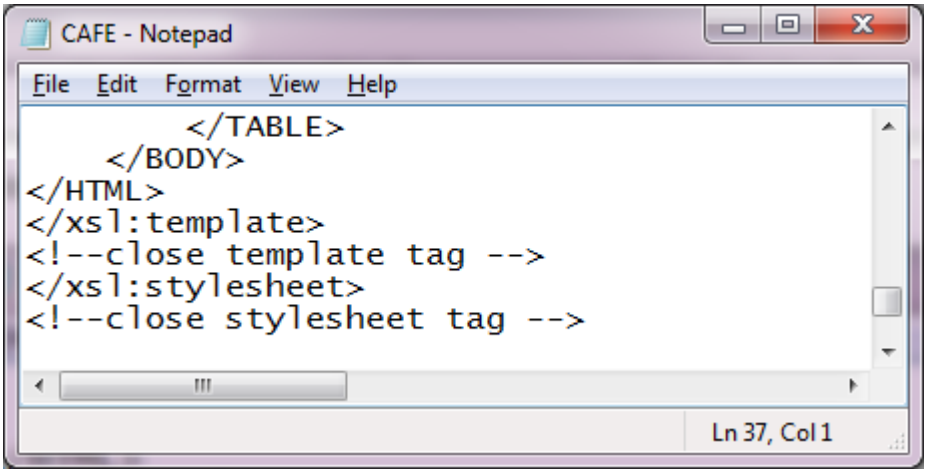
10. The header row of the table is defined by the code:



11. The XML file defines the overall structure of CAFES, which consists of multiple repeated child structure elements CAFE. The file CAFE.xml has five CAFE child structures. XSL provides the necessary language constructs to extract these values and puts them in an HTML format with the appropriate formatting. The “for-each select” looping construct of XSL is used to loop through the CAFES structure, and the XSL construct “value-of select” is used to fetch the values of the individual CAFE elements, such as CAFE_NAME and FOOD_QUALITY, and includes them in the table. This “for-each-select” looping construct of XSL acts the same as a WHILE loop in a computer program. The looping construct, such as a WHILE loop, is fixed in length, irrespective of how many CAFE child structures are involved. The looping code is as follows:

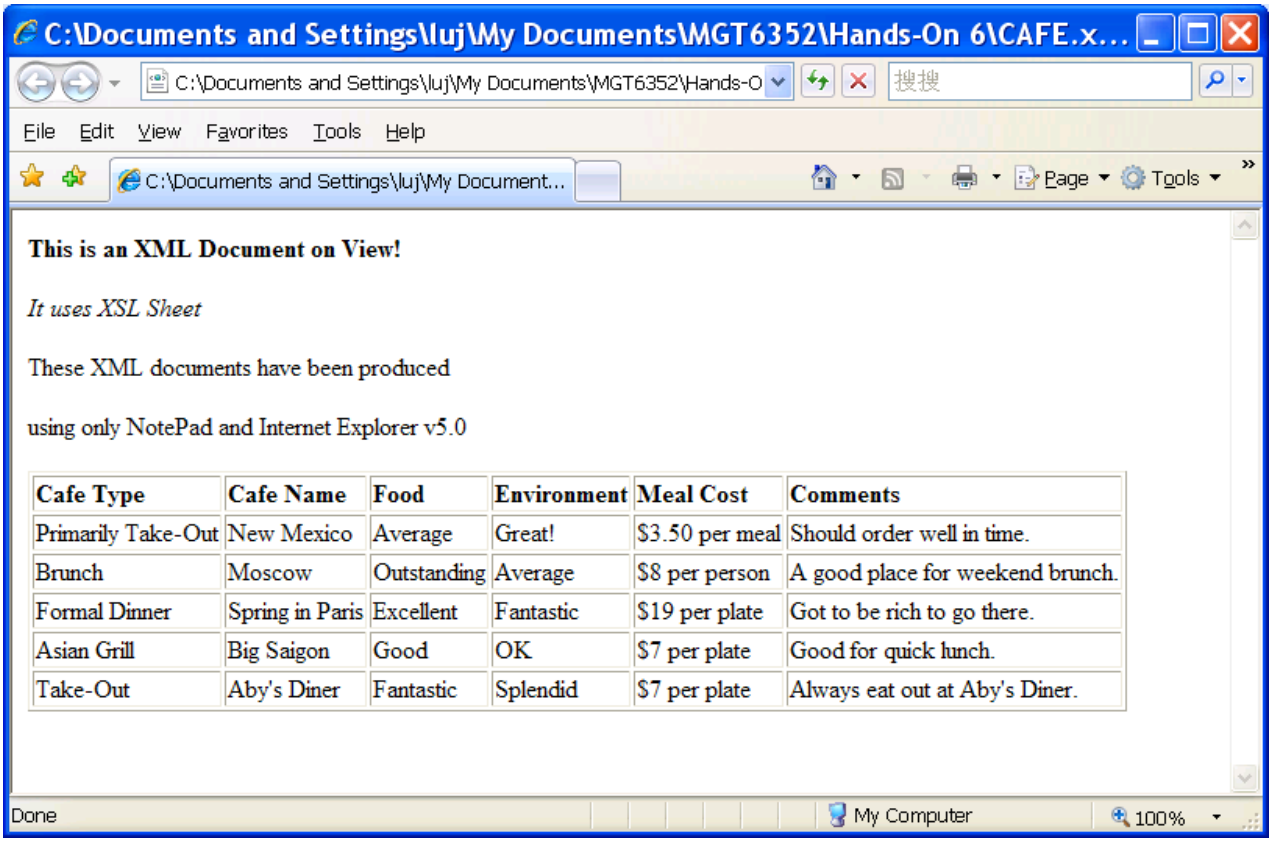


12. Now you should have keyed in the necessary code to display the comparison table. The last thing is to close up all the open tags:



13. Please remember that XML code is case sensitive. An upper cased letter is never the same as the lower cased same letter. And the select element name should be exactly the same as mentioned in your xml document. Please check both documents carefully.

14. Double-click on your xml document, and you shall see the following image on your computer screen:



3. Advanced Part (10 points)

Please add three records of your own to the xml document and add a table title to the XSL document using HTML code. Submit both files to the assignment dropbox in time.

****