XML Elements

To create an XML document, it must contain elements. Lets assume that I want to create a document with the elements LAND, FOREST, TREE, MEADOW, GRASS. Here is how I would use these elements:

```
<LAND>
    <FOREST>
        <TREE>Oak</TREE>
        <TREE>Pine</TREE>
        <TREE>Maple</TREE>
    </FOREST>
    <MEADOW>
        <GRASS>Bluegrass</GRASS>
        <GRASS>Fescue</GRASS>
        <GRASS>Rye</GRASS>
    </MEADOW>
</LAND>
```

Each element is enclosed in <> brackets. The ending element has the '/' character before its name. As you can see, there is one element that contains all others, <LAND>. XML requires one element that contains all others. This single element, which in this case is "LAND", is called th**e root element**. The FOREST element contains several TREE elements, and the MEADOW likewise contains several elements of GRASS. Each element that is contained in another ends in that same element and therefore each element is properly nested.

Elements that are included in another element are considered nested. The TREE elements in the above example are nested in the FOREST element. The FOREST element is the parent element to the TREE element and the TREE element is also called the sub-element to the FOREST element. These relationships hold true as you move up and down the element hierarchy. The FOREST and MEADOW elements are sub-elements to the LAND root element.

To understand XML terminology, you have to know how relationships between XML elements are named, and how element content is described.
Imagine that this is a description of a book:

My First XML
Introduction to XML
- What is HTML
- What is XML

XML Syntax
- Elements must have a closing tag
- Elements must be properly nested

Imagine that this XML document describes the book:

```
<book>
<title>My First XML</title>
```

```
<prod id="33-657" media="paper"></prod>
<chapter>Introduction to XML
<para>What is HTML</para>
<para>What is XML</para>
</chapter>

<chapter>XML Syntax
<para>Elements must have a closing tag</para>
<para>Elements must be properly nested</para>
</chapter>
</book>
```

Book is the root element. Title, prod, and chapter are child elements of book. Book is the parent element of title, prod, and chapter. Title, prod, and chapter are siblings (or sister elements) because they have the same parent.

Types of Elements

Elements can have different content types. **An XML element is everything from (including) the element's start tag to (including) the element's end tag.** An **element can have element content, mixed content, simple content, or empty content.** An element can also have attributes.

In the example above, **book has element content**, because it contains other elements. **Chapter has mixed** content because it contains both text and other elements. **Para has simple content** (or text content) because it contains only text. Prod has empty content, because it carries no information.

In the example above only the prod element has attributes. The attribute named id has the value "33-657". The attribute named media has the value "paper".

Classification of Elements based on its content

Elements may contain:

- Nested elements
- Processing instructions
- CDATA sections - it consist special characters which is not displayed normally by the browser such as less than or greater than sign. These signs are used to enclose tags and are special characters. An example CDATA section

```
<![CDATA[
The < and > characters are displayed normally here.
]]>
```

(Details in later section)

- Normal text.
- Entity references - An entity reference is not allowed by a & sign. (Details in later section)
- Character references - Character References are not displayed normally in XML such as the (or * characters. These characters are represented as &#40 and &#42 respectively and will be presented on the browser as the less than or greater than character they represent.
- Comments - Comments are shown below and may be placed anywhere. (Details in later section).

<!-- comment not display by browser -->

[1] *XML Elements are free and Extensible:*

XML **tags are not predefined. You must "invent" your own tags.** XML allows the author to define his own tags and his own document structure. The tags in the example (like <to> and <from>) are not defined in any XML standard. These tags are "invented" by the author of the XML document.

XML documents can be extended to carry more information.
Look at the following XML NOTE example:
<note>
<to>Tove</to>
<from>Jani</from>
<body>Don't forget me this weekend!</body>
</note>
Let's imagine that we created an application that extracted the <to>, <from>, and <body> elements from the XML document to produce this output:

MESSAGE
To: Tove
From: Jani
Don't forget me this weekend!

Imagine that the author of the XML document added some extra information to it:
<note>
<date>2002-08-01</date>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
Should the application break or crash?
No. The application should still be able to find the <to>, <from>, and <body> elements in the XML document and produce the same output. XML documents are Extensible.

1) Define XML element?
2) How XML elements are classified based on its content
3) Justify the statement XML elements are free and extensible