1. Define XML. Write internal and external DTD to describe “mail” as the root element and “to”, “from”, “subject” and “message” as the child elements. [2080 Baisakh QN. 9]

Ans. Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages.

Internal DTD:

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

<!DOCTYPE mail [

<!ELEMENT mail (to, from, subject, message)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT subject (#PCDATA)>

<!ELEMENT message (#PCDATA)>

]>

<mail>

<to>receiver @example.com</to>

<from>sender@example.com</from>

<subject>This is the subject. </subject>

<message>This is the message. </message>

</mail>

External DTD:

‘mail.dtd’ content:

<!ELEMENT mail (to, from, subject, message)>

<!ELEMENT to (#PCDATA)>

<!ELEMENT from (#PCDATA)>

<!ELEMENT subject (#PCDATA)>

<!ELEMENT message (#PCDATA)>

XML document:

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

<!DOCTYPE mail SYSTEM "mail.dtd">

<mail>

<to>recipient@example.com</to>

<from>sender@example.com</from>

<subject>This is the subject. </subject>

<message>This is the message. </message>

</mail>

1. Create an XML file to store information about a class on a module to record the student details where each student has roll\_no, name, and faculty. Display the XML using XSLT. [2079 Bhadra QN. 7]

Ans. XML file:

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

<class>

<module>Mathematics</module>

<students>

<student>

<roll\_no>101</roll\_no>

<name>John Doe</name>

<faculty>Dr. Smith</faculty>

</student>

<student>

<roll\_no>102</roll\_no>

<name>Jane Roe</name>

<faculty>Dr. Brown</faculty>

</student>

</students>

</class>

XSLT Stylesheet(student\_data.xsl):

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

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

<!-- The template matching the root element -->

<xsl:template match="/class">

<html>

<head>

<title>Student Details</title>

<style>

table {

width: 100%;

border-collapse: collapse;

}

th, td {

border: 1px solid black;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<h1>Module: <xsl:value-of select="module"/></h1>

<table>

<tr>

<th>Roll Number</th>

<th>Name</th>

<th>Faculty</th>

</tr>

<!-- Apply template for each student -->

<xsl:apply-templates select="students/student"/>

</table>

</body>

</html>

</xsl:template>

<!-- Template matching each student element -->

<xsl:template match="student">

<tr>

<td><xsl:value-of select="roll\_no"/></td>

<td><xsl:value-of select="name"/></td>

<td><xsl:value-of select="faculty"/></td>

</tr>

</xsl:template>

</xsl:stylesheet>

1. Define XML and XSLT. Explain with appropriate examples. [2078 Bhadra QN. 7]

Ans. XML is a markup language designed to store and transport data in a structured, readable format. It uses tags to define data elements and attributes, making the data both human-readable and machine-readable. XML is a flexible way to create information formats and electronically share structured data via the internet.

Example:

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

<library>

<book>

<title>The Great Gatsby</title>

<author>F. Scott Fitzgerald</author>

<year>1925</year>

<genre>Fiction</genre>

</book>

<book>

<title>1984</title>

<author>George Orwell</author>

<year>1949</year>

<genre>Dystopian</genre>

</book>

</library>

XSLT is a language for transforming XML documents into other formats such as HTML, plain text, or another XML format. XSLT uses XSL (eXtensible Stylesheet Language) to define how XML data should be transformed and presented. It separates the data from its presentation.

Example:

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

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

<!-- The template matching the root element -->

<xsl:template match="/library">

<html>

<head>

<title>Library Catalog</title>

<style>

table {

width: 100%;

border-collapse: collapse;

}

th, td {

border: 1px solid black;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<h1>Library Catalog</h1>

<table>

<tr>

<th>Title</th>

<th>Author</th>

<th>Year</th>

<th>Genre</th>

</tr>

<!-- Apply template for each book -->

<xsl:apply-templates select="book"/>

</table>

</body>

</html>

</xsl:template>

<!-- Template matching each book element -->

<xsl:template match="book">

<tr>

<td><xsl:value-of select="title"/></td>

<td><xsl:value-of select="author"/></td>

<td><xsl:value-of select="year"/></td>

<td><xsl:value-of select="genre"/></td>

</tr>

</xsl:template>

</xsl:stylesheet>

1. Define XML. What is DTD (Document Type Definition)? Design a well-formed XML document for “Teacher”. Teacher should have data such as Name, Address, Phone, Age, and Email. Name should contain FirstName and LastName. [2076 Chaitra QN. 2]

Ans. XML is a markup language designed to store and transport data in a structured, readable format. It uses tags to define data elements and attributes, making the data both human-readable and machine-readable. XML is a flexible way to create information formats and electronically share structured data via the internet.

**DTD (Document Type Definition)** is a set of rules that defines the structure and the legal elements and attributes of an XML document. It provides a way to specify the document structure and the constraints on the elements and attributes within an XML document. DTDs help ensure that XML documents are valid and conform to a predefined structure.

XML Document:

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

<teacher>

<name>

<firstname>John</firstname>

<lastname>Doe</lastname>

</name>

<address>123 Elm Street, Springfield, IL</address>

<phone>555-1234</phone>

<age>45</age>

<email>john.doe@example.com</email>

</teacher>

1. What do you mean by a well-formed XML document? Explain DTD with an example. [2074 Chaitra QN. 7]

Ans. A **well-formed XML document** adheres to the basic syntax rules defined by XML. These rules ensure that the document is readable and can be parsed correctly by XML processors. A well-formed XML document must meet the following criteria:

* **Proper Declaration**: It must start with an XML declaration (optional but recommended) that specifies the XML version and character encoding, such as <?xml version="1.0" encoding="UTF-8"?>.
* **Root Element**: It must have a single root element that encloses all other elements. This root element represents the top-level container for the document’s data.
* **Element Nesting**: Elements must be properly nested. An opening tag must have a corresponding closing tag, and elements must be closed in the reverse order of their opening.
* **Tag Matching**: Tags must be correctly matched with opening and closing pairs. For example, <tag>content</tag> is correct, while <tag>content</tag1> is not.
* **Case Sensitivity**: XML is case-sensitive, so <Tag> and <tag> are considered different. Element names and attributes must be consistently cased.
* **Attribute Quotation**: Attribute values must be enclosed in quotes (either single ' or double "). For example, <element attribute="value"/> is correct, while <element attribute=value/> is not.
* **No Overlapping Tags**: Tags cannot overlap. For instance, <tag1><tag2></tag1></tag2> is invalid because <tag1> is closed before <tag2>.
* **Special Characters**: Certain characters such as <, >, and & must be represented using entity references (&lt;, &gt;, &amp;) within element content and attribute values.

**DTD (Document Type Definition)** is a set of rules used to define the structure, elements, and attributes of an XML document. It specifies the legal building blocks of an XML document, ensuring that the document adheres to a particular format and structure. DTDs can be internal (embedded within the XML document) or external (stored in a separate file and referenced by the XML document).

Here is an example of Internal DTD:

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

<!DOCTYPE catalog [

<!ELEMENT catalog (book+)>

<!ELEMENT book (title, author, year)>

<!ELEMENT title (#PCDATA)>

<!ELEMENT author (#PCDATA)>

<!ELEMENT year (#PCDATA)>

]>

<catalog>

<book>

<title>The Great Gatsby</title>

<author>F. Scott Fitzgerald</author>

<year>1925</year>

</book>

<book>

<title>1984</title>

<author>George Orwell</author>

<year>1949</year>

</book>

</catalog>

1. Explain about XML document. Explain XSLT and WSDL.

Ans. An **XML (eXtensible Markup Language) document** is a structured text file that contains data organized in a hierarchical format. XML is designed to store and transport data, and it is both human-readable and machine-readable. Unlike HTML, which is used for displaying data, XML focuses on the structure and transport of data.

Example:

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

<catalog>

<book id="bk101">

<title>XML Developer's Guide</title>

<author>Gambardella, Matthew</author>

<genre>Computer</genre>

<price>44.95</price>

<publish\_date>2000-10-01</publish\_date>

<description>An in-depth look at creating applications with XML.</description>

</book>

<book id="bk102">

<title>Midnight Rain</title>

<author>Ralls, Kim</author>

<genre>Fantasy</genre>

<price>5.95</price>

<publish\_date>2000-12-16</publish\_date>

<description>A former architect battles corporate zombies, an evil sorceress, and her own childhood to become queen of the world.</description>

</book>

</catalog>

**XSLT** is a language used for transforming XML documents into other formats such as HTML, plain text, or another XML document. It is part of the XSL (eXtensible Stylesheet Language) family and is primarily used to apply a style sheet to XML documents.

Example:

<?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>Book Catalog</h2>

<table border="1">

<tr bgcolor="#9acd32">

<th>Title</th>

<th>Author</th>

<th>Price</th>

</tr>

<xsl:for-each select="catalog/book">

<tr>

<td><xsl:value-of select="title"/></td>

<td><xsl:value-of select="author"/></td>

<td><xsl:value-of select="price"/></td>

</tr>

</xsl:for-each>

</table>

</body>

</html>

</xsl:template>

</xsl:stylesheet>

**WSDL** is an XML-based language used for describing the functionality offered by a web service. WSDL provides a way for web services to describe their operations, parameters, and data types in a machine-readable format.

Here is a simple WSDL document for a web service that provides weather information:

Example:

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

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:tns="http://example.com/weather"

name="WeatherService"

targetNamespace="http://example.com/weather">

<types>

<xsd:schema targetNamespace="http://example.com/weather">

<xsd:element name="getWeatherRequest" type="xsd:string"/>

<xsd:element name="getWeatherResponse" type="xsd:string"/>

</xsd:schema>

</types>

<message name="getWeatherRequest">

<part name="city" element="tns:getWeatherRequest"/>

</message>

<message name="getWeatherResponse">

<part name="result" element="tns:getWeatherResponse"/>

</message>

<portType name="WeatherPortType">

<operation name="getWeather">

<input message="tns:getWeatherRequest"/>

<output message="tns:getWeatherResponse"/>

</operation>

</portType>

<binding name="WeatherBinding" type="tns:WeatherPortType">

<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="getWeather">

<soap:operation soapAction="http://example.com/weather/getWeather"/>

<input>

<soap:body use="literal"/>

</input>

<output>

<soap:body use="literal"/>

</output>

</operation>

</binding>

<service name="WeatherService">

<documentation>Weather Service</documentation>

<port name="WeatherPort" binding="tns:WeatherBinding">

<soap:address location="http://example.com/weather"/>

</port>

</service>

</definitions>