Question Bank – XML (Solved/Unsolved)

Q.1 Fill in the Blanks: (1 Mark each)

1. With XML, you can create your own __**elements**__, also called tags.
2. The beginning or first element in XML is called the __**root (document)**__ element.
3. Jon Bosak is known as the __**father**__ of XML.
4. HTML is an application of __**SGML** _.
5. The XML linking language is called __**XLink**__.
6. The CSS property __**font-weight**__ allows you to control text boldness.
7. A child element has a direct relationship to a __**parent**__ element.
8. A __**[prefix with]**__ colon in an element or attribute name must be associated with a namespace identifier.
9. An __**attribute**__ modifies an element by associating information with it.
10. Element names are __**case**__ sensitive.
11. External DTDs can reference both SYSTEM and __**PUBLIC**__ identifiers.
12. A valid XML document has a __**DTD**__ associated with it.
13. An XML document can have both an __**internal**__ and external subset.
14. <!ATTLIST ...> declares one or more __**attributes**__.
15. A content __**model**__ defines what elements may be contained within another element.
16. Element names must begin with a letter or an __**underscore**__.
17. An __**empty**__ element contains no content.
18. A __**?**__ repetition operator mean zero or one instance of the element.
19. Content models are defined with either a __**sequence**__ list or a choice list.
20. In the functional notation rgb( ), you can use numbers or __**percentages**__
21. Attribute names must begin with a __**letter**__ or underscore.
22. The keyword for an optional attribute is __**#IMPLIED**__ .
23. The __**ID**__ attribute type defines an attribute value as a unique identifier.
24. The __**xml:lang**__ attribute is a built-in XML attribute for specifying languages.
25. A __**NOTATION**__ declaration is needed for this type of enumerated attribute..
26. An internal entity is declared locally in a DTD while an __**external**__ entity is declared in a separate document.
27. A __**parameter**__ entity is legal only in a DTD.
28. A general entity reference begins with an __**ampersand (&)**__ and ends with a semicolon.
29. Any parsed entity consists of legal __**XML**__ text.
30. One tool you can use to test a DTD is with a __**conditional**__ section.
31. A namespace definition without a prefix is known as a __**default**__ namespace.
32. Namespaces are declared with an __**xmlns declaration**__ in an element start-tag.
33. The namespace __**myth**__ refers to a belief that namespaces are associated with, or validated by, schemas.
34. You can declare __**multiple**__ namespaces with multiple instances of the xmlns __**declaration**__ within a start-tag.
35. Namespaces are often declared within the start-tag of a __**root**__ element.
36. Unnamed definitions of simple or complex types are known as __**anonymous**__.
37. With <complexType>, you can define content of both __**complex**__ and mixed type.
38. minOccurs specifies the minimum times an element may occur while __**maxOccurs**__ determines the maximum times it may occur
39. The __**ref**__ attribute can reference other element and attribute definitions in a schema.

40. The <attribute> element must be declared with either a named or anonymous **__complexType__**
41. The original XML document or byte stream is referred to as a source tree and the output is called the **__result__** tree.
42. Location paths can be either abbreviated or **__unabbreviated__**.
43. A synonym for <xsl:stylesheet> is **__<xsl:transform>__**.
44. The instruction <xsl:element> is an example of an **__XSLT__** element.
45. Both <xsl:if> and <xsl:choose> help perform **__conditional__** processing.
46. The root element for XHTML is **__html__**.
47. The root element must contain a **__namespace__** declaration.
48. In XHTML, always use a **__CDATA__** section inside the <script> element.
49. The forthcoming recommendation for small devices is called XHTML **__Basic__**.
50. **__Document__** profiles defines the elements, etc. that are appropriate for a certain class of document, without a formal recommendation.

Q.2 Select True or False: (1 Mark each)

1. XML will replace HTML as the leading language for the Web. **(False)**
2. To use XML you must pay a small license fee to Sun Microsystems. **(False)**
3. A URL is a subset of the URI naming scheme. **(True)**
4. Namespaces in XML cause regrettable naming collisions. **(False)**
5. Every XML document should have a prolog or XML declaration. **(False)**
6. XML forms comments differently than SGML and HTML. **(False)**
7. HTML elements must always properly nest. **(False)**
8. Valid XML must also be well formed. **(True)**
9. It is permissible, but not mandatory, to quote XML attribute values. **(False)**
10. An internal subset requires a SYSTEM identifier. **(False)**
11. An external subset requires a URI. **(True)**
12. A validity error is always fatal. **(False)**
13. XML does not necessarily have to be well-formed, but it must be valid. **(False)**
14. An element name can begin with any character that is legal in an element. **(False)**
15. Child element and mixed content models must be enclosed in quotes. **(False)**
16. A semicolon delimits element names in a sequence list. **(False)**
17. A hexadecimal value representing an RGB triplet can be expressed in three or six digits. **(True)**
18. You can skip quotation marks around an attribute value. **(False)**
19. If you reuse a unique ID, it should generate a validity error. **(True)**
20. An unparsed entity is a non-XML data type. **(True)**
21. Certain attributes are permissible in end-tags. **(False)**
22. CSS/2's attribute selector is fully implemented in the Netscape and Microsoft browsers. **(False)**
23. An unparsed entity may require a helper application to render it. **(True)**
24. Predefined entities represent special markup characters. **(True)**
25. Parameter entity references begin with an ampersand. **(False)**
26. Unicode characters can be represented by hexadecimal numbers. **(True)**
27. Conditional sections are legal in XML documents as well as DTDs. **(False)**
28. Namespaces can be associated with schemas by a URI. **(True)**
29. A URI can be either a URL or a URN. **(True)**
30. Default namespaces apply to attributes. **(False)**
31. XML namespaces are not associated with objects. **(True)**

32. <choice> is used for grouping elements. **(True)**
33. <string> is a built-in simple datatype. **(True)**
34. A valid value for maxOccurs is unbounded. **(True)**
35. A DTD has richer datatypes than XML Schema. **(False)**
36. XML Schema unfortunately did not inherit the attribute types from XML 1.0. **(False)**
37. Location paths come from the XPath standard. **(True)**
38. You cannot embed an XSL stylesheet as you can CSS in HTML. **(False)**
39. XSLFO does not completely replace CSS. **(True)**
40. The <xsl:output> instruction is required in every XSL stylesheet. **(False)**
41. XSLT is a W3C recommendation while XSLFO is not (yet). **(True)**
42. Attribute values must be surrounded by double quotes. **(False)**
43. XHTML is moving towards modularization. **(True)**
44. It is good practice to nest <a> elements within <a> elements. **(False)**
45. Do not include the XHTML namespace when including XHTML in an XML document. **(False)**
46. The content or MIME type for XHTML is text/xhtml. **(False)**

Q.3 Select the correct answer: (1 Mark each)

1. What does XML stand for?
    1. eXtra Modern Link
    2. eXtensible Markup Language
    3. Example Markup Language
    4. X-Markup Language

    Level: Easy

2. What is the correct syntax of the declaration which defines the XML version?:
    1. <xml version="1.0" />
    2. <?xml version="1.0"?>
    3. <?xml version="1.0" />
    4. None of the above

    Level: Easy

3. Which statement is true?
    1. All the statements are true
    2. All XML elements must have a closing tag
    3. All XML elements must be lower case
    4. All XML documents must have a DTD

    Level: Easy

4. Is it easier to process XML than HTML?
    1. Yes
    2. No
    3. Somtimes
    4. Cant say

    Level: Easy

5. Which of the following programs support XML or XML applications?:

1. Internet Explorer 5.5
2. Netscape 4.7
3. RealPlayer.
4. **both 1 and 2**

Level: Easy

6. Kind of Parsers are
   1. well-formed
   2. well-documented
   3. **non-validating and validating**
   4. none of the above

   Level: Easy

7. Well formed XML document means
   1. it contains a root element
   2. it contain an element
   3. it contains one or more elements
   4. **must contain one or more elements and root element must contain all other elements**

   Level: Easy

8. Comment in XML document is given by
   1. <?-- -->
   2. <!-- --!>
   3. **<!-- -->**
   4. </-- -- >

   Level: Easy

9. When processing an output XML, "new line" symbols
   1. are copied into output "as is", i.e. "CR+LF" for Windows, CR for Macintosh, LF for Unix.
   2. **are converted to single LF symbol**
   3. are converted to single CR symbol
   4. are discarded

   Level: Easy

10. Which of the following strings are a correct XML name?
    1. **_myElement**
    2. my Element
    3. #myElement
    4. None of the above

    Level: Easy

11. Which of the following strings are a correct XML name?
    1. xmlExtension
    2. **xslNewElement**

3. XMLElement#123
4. All

Level: Easy

12. Which of the following XML fragments are well-formed?
    1. <?xml?>
    2. **<?xml version="1.0"?>**
    3. <?xml encoding="JIS"?>
    4. <?xml encoding="JIS" version="1.0"?>

    Level: Easy

13. What are the predefined attributes
    1. xml:lang
    2. xml:space
    3. **both**
    4. none.

    Level: Easy

14. Kind of Parsers are
    1. well-formed
    2. validating
    3. non-validating
    4. **Both 2 & 3**

    Level: Easy

15. Valid XML document means (most appropriate)
    (1) the document has root element
    (2) the document contains atleast one or more root element
    (3) **the XML document has DTD associated with it & it complies with that DTD**
    (4) Each element must nest inside any enclosing element property

16. XML uses the features of
    (1) HTML
    (2) XHTML
    (3) VML
    (4) SGML

    Level: Easy

17. XML document can be viewed in
    (1) IE 3.0
    (2) IE 2.0
    (3) IE 6.0
    (4) IE X.0

    Level: Medium

Topic: DTD

18. There is a way of describing XML data, how?
    1. XML uses a DTD to describe the data
    2. XML uses XSL to describe data
    3. XML uses a description node to describe data
    4. **Both 1and 3**

    Level: Medium

19. What does DTD stand for?
    1. Direct Type Definition
    2. **Document Type Definition**
    3. Do The Dance
    4. Dynamic Type Definition

    Level: Medium

20. DTD includes the specifications about the markup that can be used within the document, the specifications consists of all EXCEPT
    1. **the browser name**
    2. the size of element name
    3. entity declarations
    4. element declarations

    Level: Medium

21. Which of the following XML documents are well-formed?
    1. **<firstElement>some text goes here**
       <secondElement>another text goes here</secondElement>
       </firstElement>
    2. <firstElement>some text goes here</firstElement>
       <secondElement> another text goes here</secondElement>
    3. <firstElement>some text goes here
       <secondElement> another text goes here</firstElement>
       </secondElement>
    4. </firstElement>some text goes here
       </secondElement>another text goes here<secondElement>
       <firstElement>

    Level: Medium

22. Which of the following XML fragments are well-formed?
    1. **<myElement myAttribute="someValue"/>**
    2. <myElement myAttribute=someValue/>
    3. <myElement myAttribute='someValue'>
    4. <myElement myAttribute="someValue'/>

    Level: Medium

23. How can we make attributes have multiple values:
    1. <myElement myAttribute="value1 value2"/>
    2. <myElement myAttribute="value1" myAttribute="value2"/>

3. <myElement myAttribute="value1, value2"/>
4. **attributes cannot have multiple values**

Level: Medium

24. Which of the following XML fragments are well-formed?
    1. <myElement myAttribute="value1 <= value2"/>
    2. <myElement myAttribute="value1 & value2"/>
    3. **<myElement myAttribute="value1 > value2"/>**
    4. None of the above

    Level: Medium

25. The use of a DTD in XML development is:
    1. **required when validating XML documents**
    2. no longer necessary after the XML editor has been customized
    3. used to direct conversion using an XSLT processor
    4. a good guide to populating a templates to be filled in when generating an XML document automatically

    Level: Medium

26. Parameter entities can appear in
    1. xml file
    2. **dtd file**
    3. xsl file
    4. Both 1 and 2

    Level: Medium

27. Attribute standalone="no" should be included in XML declaration if a document:
    1. is linked to an external XSL stylesheet
    2. has external general references
    3. has processing instructions
    4. **has an external DTD**

    Level: Medium

28. In XML

    (1) the internal DTD subset is read before the external DTD
    (2) the external DTD subset is read before the internal DTD
    (3) there is no external type of DTD
    (4) there is no internal type of DTD

    Level Easy

29. Disadvantages of DTD are

    (i)DTDs are not extensible
    (ii)DTDs are not in to support for namespaces
    (iii)there is no provision for inheritance from one DTDs to another

    (1) (i) is correct
    (2) (i),(ii) are correct
    (3) (ii),(iii) are correct

(4) (i),(ii),(iii) are correct
Level: Medium

30. To use the external DTD we have the syntax
(1) <?xml version=”1.0” standalone=”no”?>
<! DOCTYPE DOCUMENT SYSTEM “order.dtd”?>
(2) <?xml version=”1.0” standalone=”yes”?>
<! DOCTYPE DOCUMENT SYSTEM “order.dtd”?>
(3)<?xml version=”1.0” standalone=”no”?>
<! DOCTYPE DOCUMENT “order.dtd”?>
(4) <?xml version=”1.0” standalone=”yes”?>
<! DOCTYPE DOCUMENT SYSTEM “order.dtd”?>
Level: Medium

31. To add the attribute named Type to the <customer> tag the syntax will be
(1) <customer attribute Type=”exelent”>
(2) <customer Type attribute =”exelent”>
(3) <customer Type attribute_type=”exelent”>
(4) <customer Type=”exelent”>
Level: Medium

32. The syntax for parameter entity is
(1) <! ENTITY % NAME DEFINITION>
(2) < ENTITY % NAME DEFINITION>
(3) <! ENTITY $ NAME DEFINITION>
(4) < ENTITY % NAME DEFINITION>
Level: Medium

Topic: Schema

33. You can name the schema using the name attribute like
1. <schema attribute=”schema1”>
2. <schema nameattribute=”schema1”>
3. <schema nameattri=”schema1”>
4. **<schema name=”schema1”>**
Level: Medium

34. The default model for complex type, in XML schemas for element is
1. textOnly
2. **elementOnly**
3. no default type
4. both 1 & 2
Level: Medium

35. Microsoft XML Schema Data types for Hexadecimal digits representating octates
1. UID
2. UXID
3. **UUID**

4. XXID
Level: Medium

36. A schema describes
(i) grammer
(ii) vocabulary
(iii) structure
(iv) datatype of XML document
(1) (i) & (ii) are correct
(2) (i),(iii) ,(iv) are correct
(3) (i),(ii),(iv) are correct
(4) (i),(ii),(iii),(iv) are correct
Level: Medium

37. Microsoft XML Schema Data Type "boolean" has values
(1) True ,False
(2) True ,False or 1,0
(3) 1,0
(4) any number other then zero and zero
Level: Difficult

38. Simple type Built into Schema "data' represent a data in
(1) MM-DD-YY
(2) Dd-MM-YY
(3) YY-MM-DD
(4) YYYY-MM-DD
Level: Medium

39. In simple Type Built into XML schema Boolean type holds
(1) True, False
(2) 1,0
(3) both (1) & (2)
(4) True/False and any number except 0
Level: Medium

40. In simple type built into XML schema type flat has single precision of _____ floating point
(1) 16 bit
(2) 32 bit
(3) 8 bit
(4) 4 bit
Level: Medium

Topic: Misc.

41. The XML DOM object is

1. Entity
2. **Entity Reference**
3. Comment Reference
4. Comment Data

Level: Medium

42. Attribute of the document interface in DOM is/are
    (i)doctype
    (ii)implementation
    (iii)documentElement
    which are read only attributes
    (1) (i) only
    (2) (ii) only
    (3) (ii),(iii) only
    (4) all

Level: Medium

43. The default model for complex type, in XML schemas for element is
    (1) textOnly
    (2) elementOnly
    (3) no default type
    (4) both a & b
    Level: Easy

44. To create a choise in XML schemas, we use the
    (1) <xsd:select> element
    (2) <xsd:multi> element
    (3) <xsd:choise> element
    (4) <xsd:single> element
    Level: Medium

45. The XML DOM object is
    (1) Entity
    (2) Entity Reference
    (3) Comment Reference
    (4) Comment Data
    Level: Medium

46. To create a data island we use the _____HTML element
    (1) <XML>
    (2) <dataisland>
    (3) <Island>
    (4) <XMLIsland>
    Level: Medium

47. To Bind the HTML elements with DSO we use _____ attribute
    (1) DATASOURCE

(2) DATAFIELD
(3) DATASRC
(4) DATAFLD
Level: Medium

48. To bind the HTML element <INPUT> Type in text with the datasource "dsoCustomer" we use
    (1) <INPUT TYPE="TEXT" DATAFIELD="#dsoCustomer">
    (2) <INPUT TYPE="TEXT" DATASRC="dsoCustomer">
    (3) <INPUT TYPE="TEXT" DATASRC="#dsoCustomer">
    (4) <INPUT TYPE="TEXT" DATAFLD="#dsoCustomer">
    Level: Medium

49. XML DSOs has the property for the number of pages of data the recordset contains
    (1) count
    (2) number
    (3) pageCount
    (4) pageNumber
    Level: Medium

50. Whats so great about XML?
    (1) Easy data exchange
    (2) High speed on network
    (3) Only (2)is correct
    (4) Both (1) & (2)
    Level: Medium

51. For XML document to be valid
    (1) document need to be well formed also
    (2) document need not to be well formed
    (3) document need to be well formed & valid
    (4) document validity has no relationship with well formedness
    Level: Medium

52. A textual object is a well formed XML document if
    (i) Taken as a whole it matches the production labeled document.
    (ii) Each of the parsed entity which is referenced directly or indirectly within the document can be well formed
    (1) (i) is correct
    (2) (ii)is correct
    (3) both are correct
    Level: Medium

53. <?xml version="1.0" standalone="yes" encoding="UTF-8"?>
    (1) it shows that the version is 1.0
    (2) shows thatit is standalone

(3) the standalone is wrong
(4) version attribute is not in XML
Level: Medium

54. The attribute used to define a new namespace is
(1) XMLNS
(2) XmlNameSpace
(3) Xmlns
(4) XmlNs
Level: Medium

Topic: Templates

55. To match the root node in XMLT transform the syntax will be
1. <xsl:template match="Document">
2. <xsl:template match="Root">
3. <xsl:template match="RootNode">
4. <xsl:template match="/">
Level: Medium

56. To match the specific XML elements child like <NAME> of parent element is <PLANET> the syntax will be
1. <xsl:template match="PLANET_NAME">
2. <xsl:template match="PLANET/NAME">
3. <xsl:template match="/NAME">
4. <xsl:template match="//">
Level: Medium

57. PI in XML specification stands for
1. 3.14
2. priceless instruction
3. processing instruction
4. polymorphic inheritance
Level: Medium

58. A validating XML application should be used when:
1. the design demands that all elements use both start and end tags
2. missing or out-of-place elements could cause application errors
3. attribute values cannot refer to external entity references
4. High performance is an important architectural constraint
Level: Medium

59. A DSO operates like
(a) data simulation object at server side
(b) dynamic source object at client side
(c) data source object at client side
(d) data simulation object at client side

Ans: (c)

Topic: XSL

60. The XSL formating object use to format a list is
    1. **list-block**
    2. list-item
    3. list-item-body
    4. list-item-label

    Level: Difficult

61. The attribute used to define a new namespace is
    1. XMLNS
    2. XmlNameSpace
    3. **Xmlns**
    4. XmlNs

    Level: Difficult

---

62. Identify the most accurate statement about the application of XML:
    1. XML must be used to produce XML and HTML output.
    2. XML cannot specify or contain presentation information.
    3. **XML is used to describe hierarchically organized information.**
    4. XML performs the conversion of information between different e-business applications.

    Level: Difficult

63. The XSl formatting object which formats the data and caption of a table is
    (1) table
    (2) table-content
    (3) table-text
    (4) none of the above

    Level: Difficult

64. The XSL formating object which holds the content of the table body
    (1) table
    (2) table-body
    (3) table-content
    (4) table-footer

    Level: Difficult

65. The XSL formatting object which formats the data in a table
    (1) table
    (2) table-body
    (3) title
    (4) table-content

    Level: Difficult

66. The XSL formating object use to hold the content of the label of a list item is
    (1) list-block
    (2) list item
    (3) list-item-body
    (4) list-item-label
    Level: Difficult

67. The XSL formating object use to hold the contents of the body of a list item is
    (1) list-block
    (2) list item
    (3) list-item-body
    (4) list-item-label
    Level: Difficult

68. XSL has formatting object "block"
    (1) is not supported in XSL
    (2) generates a block level reference area
    (3) create a display block
    (4) groups global declarations for a style sheet
    Level: Difficult

69. XSL has "block container" for formating the document
    (1) to create a display block to format the titles
    (2) to create a display block to format the paragraphes
    (3) to create a display block to format the headlines & figures
    (4) to create a block level reference area
    Level: Difficult

70. The syntax for writing the minimum occurrence for an element is
    (1) <xsd:element ref="note" min="0"/>
    (2) <xsd:elements ref="note" min="0"/>
    (3) <xsd:elements ref="note" minOccur="0"/>
    (4) <xsd:elements ref="note" minOccurs="0"/>
    Level: Medium

71. The syntax for writing default values for element is
    (1) <xsd:element name="max" type="xsd:integer" value="100"/>
    (2) <xsd:element name="max" type="xsd:integer" fixValue="100"/>
    (3) <xsd:element name="max" type="xsd:integer" default="100"/>
    (4) <xsd:element name="max" type="xsd:integer" defaultval="100"/>

Topic: XSLT , X-Pointers, XML

72. To use XSLT in an XML system:
    1. the input and output of the XSLT processor must be unparsed XML documents

2. **the input and output of the XSLT processor must be a hierarchical tree representing an XML document**
3. the XSLT processor must be called from a web agent
4. the XSLT processor must be given the DTD as well as the XML document instance

Level: Difficult

73. What is the role of the XPath language in XSL processing?
    1. XPath identifies the order or path of processing to be followed as the XSL language is processed
    2. **XPath identifies locations in XML data to be transformed in the source tree and the locations to be generated in output tree specified in XSL translation prescriptions**
    3. XPath identifies the path to be followed in the execution of XSL translation prescriptions
    4. XPath specifies which XSL transform files are to be used in the translation of XML

    Level: Difficult

74. Which statement correctly describes the capabilities of the XSLT language?
    1. XSLT uses the DTD to determine how XML documents will be translated
    2. XSLT specifies how a hierarchical trees, representable by an XML document may be translated into non-hierarchical formats
    3. **XSLT specifies how a hierarchical tree, representable by an XML document, may be translated into another hierarchical tree, also representable by an XML document**
    4. XSLT specifies the formatting style to be used to render an XML document

    Level: Difficult

75. XSLT processors accept as input:
    1. **an XML conforming document file and an XSLT specification file**
    2. only an XML document
    3. only an XSLT specification
    4. either an XML document or an XSLT specification

    Level: Difficult

76. The transformation of XML document in to another type of document by XSLT can be done by
    (i)In the server
    (ii)In the client
    (iii)With a separate program

    (1) only(i) & (ii)
    (2) only (ii) & (iii)
    (3) all are correct

(4) only (i) & (iii)
Level: Difficult

77: To match the root node in XMLT transform the syntax will be
(1) <xsl:template match="Document">
(2) <xsl:template match="Root">
(3) <xsl:template match="RootNode">
(4) <xsl:template match="/">
Level: Difficult

78: To match the specific XML elements in XMLT the syntax for given name "rootnode" is
(1) <xsl:template match="root">
(2) <xsl:template match="/">
(3) <xsl:template match="rootnode">
(4) <xsl:template match="//">
Level: Difficult

79 To match the specific XML elements child like <NAME> of parent element is <PLANET> the syntax will be
(1) <xsl:template match="PLANET_NAME">
(2) <xsl:template match="PLANET/NAME">
(3) <xsl:template match="/NAME">
(4) <xsl:template match="//">
Level: Difficult

80. InXSLT style sheet we have syntax to match elements with id as (if id is "change")
(1) <xsl:template match="id('change')">
(2) <xsl:template match="(change)">
(3) <xsl:template match="change">
(4) <xsl:template match-id="Change">
Level: Difficult

81. To match the text node (in XSLT) the syntax will be
(1) <xsl:template match="text">
(2) <xsl:template match-text="text">
(3) <xsl:template match=text( )>
(4) <xsl:template match="text( )">
Level: Difficult

82. An element declaration specifies
1. **a single markup element**
2. zmarkup elements
3. markup data
4. the document data

Level: Easy

83. Well formed XML document means(most appropriate)
    1. it contains a root element
    2. it contain an element
    3. it contains one or more elements
    4. must contain one or more elements and root element must contain all other elements

    Level: Easy

84: Which of the following specify that the order and content of "membership" is not important
1. <!ELEMENT membership NORULE>
2. <!ELEMENT membership EMPTY>
3. <!ELEMENT membership ALL>
4. <!ELEMENT membership ANY>

Level: Easy

85: Which of the following is used to specify the attribute list of an element
1. ATTLIST
2. ?ATTLIST
3. !ATTLIST
4. #ATTLIST

Level: Medium

86: Which of the following instruct the browser which stylesheet to use
1. <xml-stylesheet type="text/xsl" href="cd.xsl">
2. <xml-stylesheet type="text/xsl" xsl="cd.xsl">
3. <?xml-stylesheet type="text/xsl" href="cd.xsl"?>
4. <?xml-stylesheet type="text/xsl" xsl="cd.xsl"?>

Level: Difficult

88: Which of the following XSLT Patterns is used to match any descendant nodes
1) /
2) //
3) .
4) ..
Level: Medium

89: Which of the following XSLT Patterns is used to match the parent node
1) /
2) //
3) .
4) ..
Level: Medium

90: Which of the following is a valid XSLT iteration command
1) for
2) for-all
3) for-each
4) in-turn
Level: Medium

91.What is an advantage of XML compared to HTML?
1) XML works on more platforms.
2) **XML is suited to using Web pages as front ends to databases.**
3) XML was designed for portable phones.
4) XML is simpler to learn than HTML.
Level: Difficult

92.The following best describes the development of XML.
1. XML developed from HTML because WEB browsers became more powerful.
2. XML is designed as a replacement because SGML can not be used for document development.
3. XML builds on HTMLs ability to provide content to virtually any audience by adding the power of intelligent content.
4. XML is the modern replacement for HTML and SGML, taking the good points from each, making both of those languages obsolete.
Level: Medium

93) The correct priority for implementing XML based IETMs is :
1. Develop DTD, conduct a pilot project, create a modular library, train staff.
2. Train staff, convert legacy documents, develop DTD, create modular library.
3. Conduct pilot program, train staff, create modular library, develop DTD
4. Conduct pilot program, train staff, develop DTD, convert documents, purchace XML tools.
Level: Difficult

94. Which of the following statements is true:
1. XML is a direct subset of SGML
2. SGML is an application of HTML
3. XML is a kind of dynamic HTML
4. XHTML is XML rewritten in HTML
5. SGML and XML are the same thing
Level: Difficult

95. What is a qualified name?
1. Any name conforming to the XML Names specification
2. A name having prefix and local name separated by a colon
3. A name applying only to qualified elements and attributes
4. None of the above
Level: Dfficult

96. What is a NCName
1. A Non-Common Name
2. A Non-Conforming Name
3. **A Non-Colonized Name**
4. None of the above

Level:Difficult

97. If a namespace is attached to an element by prefix, what is the effect on non-prefixed child elements
1. Nothing
2. The namespace affects the immediate nonprefixed child elements, but no others
3. The namespace affects all child elements of the element to which the namespace is attached no matter what level.
4. None of the above

Level:Difficult

98. What is the default namespace
1. The namespace used by default when no namespace is declared
2. The namespace used when two or more namespaces are referenced
3. **A namespace that is referenced with the xmlns attribute, but without a prefix**
4. None of the above

Level:Difficult

99.What is an XML namespace?
1. A set of names applied to specific spaces within an XML document, such as the head and body
2. **A set of names representing a specific XML vocabulary**
3. A set of names for XML documents pertaining to a particular vocabulary
4. None of the above.

Level:Difficult

100. From what set of names do NCNames derive?
1. Any combination of characters allowable in XML
2. **Any names conforming to XML Names, minus the colon**
3. Any names for elements and attributes within the DTD to which the namespace refers
4. None of the above.

Level:Difficult

Q.4. Match the following

Question Bank – XML (Solved/Unsolved)

1.

c	1. Inventor of the Web and HTML	a.	James Clark
d	2. Marries HTML and XML	b.	Goldfarb
e	3. First line in an XML document	c.	Berners-Lee
a	4. Came up with the name XML	d.	XHTML
b	5. GML and SGML author	e.	XML declaration

2.

d	1. Generated if XML is *not* well formed	a.	selector
e	2. Value for `display` property	b.	points
f	3. Element that contains no content	c.	DTD
b	4. 72 in an inch	d.	fatal error
a	5. Indicates element to apply style to	e.	`inline`
c	6. Document Type Definition	f.	empty element

3.

d	1. Document type declaration	a.	PCDATA
e	2. Children of a parent element	b.	CDATA section
f	3. Element declaration	c.	`<`
c	4. Built-in XML entity	d.	`<!DOCTYPE…`
a	5. Parsed character data	e.	siblings

b	6. Hides markup from an XML processor	a. `<!ELEMENT…`

4.

C	1. CSS `visibility` property	a. base16
d	2. generic font name	b. `rgb(0,0,0)`
f	3. zero or more instance repetition operator	c. `hidden`
b	4. functional notation	d. `sans-serif`
a	5. Hexadecimal	e. or
e	6. Vertical bar (`|`)	f. asterisk (`*`)

5.

d	1. attribute declaration	a. `CDATA`
e	2. a tokenized attribute type	b. `NOTATION`
a	3. string attribute type	c. `#FIXED`
c	4. attribute and default value must always be provided	d. `<!ATTLIST>`
b	5. an enumerated attribute type	e. `IDREF`

6.

d	1. unparsed entity	a. `&#x0041;`
e	2. conditional section	b. `&lt;`
b	3. built-in or predefined entity	c. `%name;`
a	4. character reference	d. Word file
C	5. parameter entity	e. `<![IGNORE[`

7.

e	1. `urn:wyeast-comm:db`	a. `#FIXED`
d	2. Can share the same name if in different elements	b. prefix
f	3. For declaring a namespace	c. URL
c	4. `http://wyeast.net/tack`	d. attribute
b	5. `db:`	e. URN
a	6. Used to declare a default in a DTD	f. `xmlns`

8.

d	1. XML Schema predecessor	a. simple type
c	2. part of an annotation	b. decimal
e	3. immediate child of `<schema>`	c. `<appinfo>`
b	4. a built-in datatype	d. DTD
a	5. no attributes, other elements as content	e. global

9.

D	1. instructions for transforming an XML document	a. `@*`
e	2. formats XML documents	b. XSLT
b	3. transforms XML documents	c. `<xsl:text>`
a	4. location path for all attribute nodes	d. template
c	5. **XSLT** element	e. XSLFO

10.

c	1. Marriage of XML and HTML	a. whitespace
e	2. Contains deprecated elements and attributes	b. XHTML namespace
d	3. All XHTML elements and attributes must be in this form	c. XHTML
a	4. Avoid in attribute values	d. lowercase
b	5. `http://www.w3.org/1999/xhtml`	e. transitional DTD

Q.4. Answers the following questions: (3 Mark each)

1) Describe the role that XSL can play when dynamically generating HTML pages from a relational database.
2) Give a few examples of types of applications that can benefit from using XML.
3) What is DOM and how does it relate to XML?
4) What is SOAP and how does it relate to XML?
5) Can you walk us through the steps necessary to parse XML documents?
6) Give some examples of XML DTDs or schemas that you have worked with.
7) Using XSLT, how would you extract a specific attribute from an element in an XML document?
8) When constructing an XML DTD, how do you create an external entity reference in an attribute value?
9) How would you build a search engine for large volumes of XML data?
10) Describe the differences between XML and HTML.