**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

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

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

4. Is it easier to process XML than HTML?

**1. Yes**

2. No

3. Somtimes

4. Cant say

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**

6. Kind of Parsers are

1. well-formed

2. well-documented

**3. non-validating and validating**

4. none of the above

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**

8. Comment in XML document is given by

1. <?-- -->

2. <!-- --!>

**3. <!-- -->**

4. </-- -- >

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

10. Which of the following strings are a correct XML name?

**1. \_myElement**

2. my Element

3. #myElement

4. None of the above

11. Which of the following strings are a correct XML name?

1. xmlExtension

**2. xslNewElement**

3. XMLElement#123

4. All

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"?>

13. What are the predefined attributes

1. xml:lang

2. xml:space

**3. both**

4. none.

14. Kind of Parsers are

1. well-formed

2. validating

3. non-validating

**4. Both 2 & 3**

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**

17. XML document can be viewed in

(1) IE 3.0

(2) IE 2.0

**(3) IE 6.0**

(4) IE X.0

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**

19. What does DTD stand for?

1. Direct Type Definition

**2. Document Type Definition**

3. Do The Dance

4. Dynamic Type Definition

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

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>

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’/>

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**

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

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

26. Parameter entities can appear in

1. xml file

**2. dtd file**

3. xsl file

4. Both 1 and 2

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**

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

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**

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”?>

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” >**

32. The syntax for parameter entity is

**(1) <! ENTITY % NAME DEFINITION>**

(2) < ENTITY % NAME DEFINITION>

(3) <! ENTITY $ NAME DEFINITION>

(4) < ENTITY % NAME DEFINITION>

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”>**

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

35. Microsoft XML Schema Data types for Hexadecimal digits representating octates

1. UID

2. UXID

**3. UUID**

4. XXID

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**

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

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**

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

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

41. The XML DOM object is

1. Entity

**2. Entity Reference**

3. Comment Reference

4. Comment Data

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**

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

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

45. The XML DOM object is

(1) Entity

**(2) Entity Reference**

(3) Comment Reference

(4) Comment Data

46. To create a data island we use the \_\_\_\_\_\_\_\_\_\_\_\_\_HTML element

**(1) <XML>**

(2) <dataisland>

(3) <Island>

(4) <XMLIsland>

47. To Bind the HTML elements with DSO we use \_\_\_\_\_\_\_\_\_ attribute

**(1) DATASOURCE**

(2) DATAFIELD

**(3) DATASRC**

(4) DATAFLD

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”>

49. XML DSOs has the property for the number of pages of data the recordset contains

(1) count

(2) number

**(3) pageCount**

(4) pageNumber

50. Whats so great about XML?

(1) Easy data exchange

(2) High speed on network

(3) Only (2)is correct

**(4) Both (1) & (2)**

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

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**

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

54. The attribute used to define a new namespace is

(1) XMLNS

(2) XmlNameSpace

**(3) Xmlns**

(4) XmlNs

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=” /”>**

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=”//”>

57. PI in XML specification stands for

1. 3.14

2. priceless instruction

**3. processing instruction**

4. polymorphic inheritance

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

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)

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

61. The attribute used to define a new namespace is

1. XMLNS

2. XmlNameSpace

**3. Xmlns**

4. XmlNs

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.

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**

64. The XSL formating object which holds the content of the table body

(1) table

**(2) table-body**

(3) table-content

(4) table-footer

65. The XSL formatting object which formats the data in a table

**(1) table**

(2) table-body

(3) title

(4) table-content

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**

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

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

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**

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” />**

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” />

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

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

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

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

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)

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=” /” >**

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=” //”>

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”>

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( )” >**

82. An element declaration specifies

**1. a single markup element**

2. zmarkup elements

3. markup data

4. the document data

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

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>**

85: Which of the following is used to specify the attribute list of an element

1. ATTLIST

2. ?ATTLIST

**3. !ATTLIST**

4. #ATTLIST

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"?>

88: Which of the following XSLT Patterns is used to match any descendant nodes

1) /

**2) //**

3) .

4) ..

89: Which of the following XSLT Patterns is used to match the parent node

1) /

2) //

3) .

**4) ..**

90: Which of the following is a valid XSLT iteration command

1) for

2) for-all

**3) for-each**

4) in-turn

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.

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.

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.

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

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

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

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

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

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.

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.