Free Questions for I10-002 by actualtestdumps

Shared by Compton on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Select the following DOM (Level 2) methods that can add an attribute node (Attr) to an element node (Element). (Multiple answers possible. Select two.)

Options:

A- appendChild

B- hasAttribute

C- setAttribute

D- setAttributeNode

Answer:

C, D

Question 2

Question Type: MultipleChoice

Select a valid XML Document against the "XML Schema" referenced when the Exhibit Button is pushed.

[XML Schema]

Options:

A- <TestML xmlns='urn:xmlmaster:testml'>
<scenario level='1' data='100'>
<title>Prologue</title>
<content>Long long ago...</content>
</scenario>
</TestML>

B- <TestML xmlns='urn:xmlmaster:testml'>
<scenario level='1' data='100'>
<title xmlns='' >Prologue</title>
<content xmlns='' >Long long ago...</content>
</scenario>
</TestML>

C- <TestML xmlns='urn:xmlmaster:testml'
xmlns:tns='urn:xmlmaster:testml' >
<scenario tns: level='1' tns: data='100'>
<title>Prologue</title>
<content>Long long ago...</content>

</scenario>
</TestML>

D- <TestML xmlns='urn:xmlmaster:testml'
xmlns:tns='urn:xmlmaster:testml' >
<scenario tns: level='1' tns: data='100'>
<title xmlns='' >Prologue</title>
<content xmlns='' >Long long ago...</content>
</scenario>
</TestML>

Answer:

B

Question 3

Question Type: **MultipleChoice**

Push the Exhibit Button to load the referenced "XML Document". Create an XML Schema Document for "XML Document". The definitions of this XML Schema Document require that the value of the level element must be singularly unique within the XML Document. Which of the following correctly describes the XML Schema Document?

Options:

A- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:element name='TestML' type='testmlType'>
<xs:unique name='levelUnique'>
<xs:selector xpath='record/level' />
<xs:field xpath='record/level' />
</xs:unique>
</xs:element>
<xs:complexType name='testmlType'>
<xs:sequence>
<xs:element ref='record' maxOccurs='unbounded' />
</xs:sequence>
</xs:complexType>
<xs:element name='record' type='recordType' />
<xs:complexType name='recordType'>
<xs:sequence>
<xs:element ref='level' />
<xs:element ref='data' />
</xs:sequence>
</xs:complexType>
<xs:element name='level' type='xs:int' />
<xs:element name='data' type='xs:int' />
</xs:schema>

B- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>

<xs:element name='TestML' type='testmlType'>
<xs:unique name='levelUnique'>
<xs:selector xpath='record' />
<xs:field xpath='level' />
</xs:unique>
</xs:element>
<xs:complexType name='testmlType'>
<xs:sequence>
<xs:element ref='record' maxOccurs='unbounded' />
</xs:sequence>
</xs:complexType>
<xs:element name='record' type='recordType' />
<xs:complexType name='recordType'>
<xs:sequence>
<xs:element ref='level' />
<xs:element ref='data' />
</xs:sequence>
</xs:complexType>
<xs:element name='level' type='xs:int' />
<xs:element name='data' type='xs:int' />
</xs:schema>

C- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:element name='TestML' type='testmlType' />
<xs:complexType name='testmlType'>
<xs:sequence>

<xs:element ref='record' maxOccurs='unbounded' />
</xs:sequence>
</xs:complexType>
<xs:element name='record' type='recordType'>
<xs:unique name='levelUnique'>
<xs:selector xpath='level' />
<xs:field xpath='level' />
</xs:unique>
</xs:element>
<xs:complexType name='recordType'>
<xs:sequence>
<xs:element ref='level' />
<xs:element ref='data' />
</xs:sequence>
</xs:complexType>
<xs:element name='level' type='xs:int' />
<xs:element name='data' type='xs:int' />
</xs:schema>

D- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:element name='TestML' type='testmlType' />
<xs:complexType name='testmlType'>
<xs:sequence>
<xs:element ref='record' maxOccurs='unbounded' />
</xs:sequence>
</xs:complexType>

```
<xs:element name='record' type='recordType'>
<xs:unique name='levelUnique'>
<xs:selector xpath='level' />
<xs:field xpath='.' />
</xs:unique>
</xs:element>
<xs:complexType name='recordType'>
<xs:sequence>
<xs:element ref='level' />
<xs:element ref='data' />
</xs:sequence>
</xs:complexType>
<xs:element name='level' type='xs:int' />
<xs:element name='data' type='xs:int' />
</xs:schema>
```

Answer:

B

Question 4

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "testml.xsd". Assume that "testml.xsd" is defined. Without rewriting this XML Schema Document ("testml.xsd"), create a new, separate XML Schema Document to partially change the schema definition replacing the phone element with a cellPhone element. As a result, the following "XML Document" will be valid against the new schema. Which of the following correctly describes the new XML Schema Document?

Assume that the XML parser correctly processes the XML Schema schemaLocation attribute.

[XML Document]

```
<TestML>
  <person>
    <name>John Smith</name>
    <cellPhone>000-1111-2222</cellPhone>
  </person>
</TestML>
```

Options:

A- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:import schemaLocation='testml.xsd' />
<xs:element name='cellPhone' type='xs:string' />
</xs:schema>

B- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>

<xs:include schemaLocation='testml.xsd' />
<xs:element name='cellPhone' substitutionGroup='phone' type='xs:string' />
</xs:schema>

C- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:redefine schemaLocation='testml.xsd'>
<xs:element name='cellPhone' base='phone' type='xs:string' />
</xs:redefine>
</xs:schema>

D- It is not possible to implement a function of the type proposed.

Answer:

B

Question 5

Question Type: **MultipleChoice**

Which of the following correctly describes the output results when performing an XSLT tranformation on the following "XML Document" using the "XSLT Style Sheet" below? Note that the XSLT processor can output transformation results as a document.

[XML Document]

```
<data xml:lang="en">Tokyo</data>
```

[XSLT Style Sheet]

```
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:template match="/">
    <xsl:apply-templates select="data" />
 </xsl:template>
    <xsl:template match="data">
   <xsl:copy>
     <xsl:apply-templates select="." mode="processLang" />
     <xsl:value-of select="." />
   </xsl:copy>
 </xsl:template>

  <xsl:template match="*" mode="processLang">
  <xsl:choose>
    <xsl:when test="@xml:lang">
      <xsl:copy-of select="@xml:lang" />
    </xsl:when>
    <xsl:otherwise>
      <xsl:attribute name="xml:lang">ja-JP</xsl:attribute>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
```

Options:

A- <data/>

B- <data>Tokyo</data>

C- <data xml:lang='en'>Tokyo</data>

D- <data xml:lang='ja-JP'>Tokyo</data>

Answer:

C

Question 6

Question Type: **MultipleChoice**

Push the Exhibit Button to load the referenced "XSLT Style Sheet". Select which of the following correctly describes the output results of an XSLT transformation of the "XML Document" using the "XSLT Style Sheet".

[XSLT Style Sheet]

Note that the XSLT processor can output transformation results as a document. Line feeds and indents are not reflected.

[XML Document]

89

70

Options:

A- <record>
<data value='89'/>
<data value='70'/>
</record>

B- <record xmlns='urn:xmlmaster:B'>
<data value='89'/>
<data value='70'/>
</record>

C- <record xmlns='urn:xmlmaster:B'>
<data B:value='89' xmlns:B='urn:xmlmaster:B'/>
<data B:value='70' xmlns:B='urn:xmlmaster:B'/>
</record>

D- <record>
<data value='89' xmlns='urn:xmlmaster:A'/>
<data value='70' xmlns='urn:xmlmaster:A'/>

</record>

E- <record>
<data B:value='89' xmlns='urn:xmlmaster:A' xmlns:B='urn:xmlmaster:B'/>
<data B:value='70' xmlns='urn:xmlmaster:A' xmlns:B='urn:xmlmaster:B'/>
</record>

Answer:

B

Question 7

Question Type: MultipleChoice

Which of the following text strings is not output to the results document when performing an XSLT transformation on the following "XML Document" using the "XSLT Style Sheet" below? Note that the XSLT processor can output transformation results as a document. [XML Document]

Options:

A- doc

B- NS1:title

C- body

D- NS1:section

E- NS1:content

Answer:

C

Question 8

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML Document." Assume that the "XML Document" is changed to the "Results XML Document." Select which XSLT style sheet correctly performs the transformation.

[XML Document]

ABC

Note that the XSLT processor can output transformation results as a document. (Multiple answers possible. Select two.)

[Results XML Document]

AC

Options:

A- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match='/'>
<root><xsl:apply-templates select=' //test '/></root>
</xsl:template>
</xsl:stylesheet>

B- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match='/'>
<root><xsl:apply-templates select=' /test/text() '/></root>
</xsl:template>
</xsl:stylesheet>

C- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match='/'>
<root><xsl:apply-templates select=' test '/></root>
</xsl:template>
</xsl:stylesheet>

D- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match='/'>
<root><xsl:apply-templates select=' //data/ancestor::*/text() '/></root>
</xsl:template>

</xsl:stylesheet>

Answer:

B, D

Question 9

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML Document". Select which of the following correctly describes the output results of an XSLT transformation of the "XML Document" using the "XSLT Style Sheet".

[XML Document]

lmnop

Note that the XSLT processor can output transformation results as a document. Line feeds and indents are not reflected.

[XSLT Style Sheet]

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <out>
      <xsl:apply-templates select="root"/>

    </out>
  </xsl:template>
  <xsl:template match="root">
    <xsl:call-template name="namedTemplate"/>
    <xsl:apply-templates select="data" />
  </xsl:template>
  <xsl:template match="data" name="namedTemplate">
    <TEST><xsl:copy-of select="." /></TEST>
  </xsl:template>
</xsl:stylesheet>
```

Options:

A- <out>
<TEST><root><data>lmnop</data></root></TEST>
<TEST><root><data>lmnop</data></root></TEST>
</out>

B- <out>

<TEST><data>lmnop</data></TEST>
<TEST><data>lmnop</data></TEST>
</out>

C- <out>
<TEST><data>lmnop</data></TEST>
<TEST><root><data>lmnop</data></root></TEST>
</out>

D- <out>
<TEST><root><data>lmnop</data></root></TEST>
<TEST><data>lmnop</data></TEST>
</out>

Answer:

D

Question 10

Question Type: **MultipleChoice**

Push the Exhibit Button to load the referenced "XML Document 1" and "XML Document 2," and process XML using "DOM Processing".

Select which of the following is the most appropriate expression of the results under XML 1.0. Line feeds and/or indents are not reflected in the results.

[DOM Processing]

Process XML using the following method.

Document output = updateXML(doc1, doc2);

The variable doc1 here references the Document instance of the loaded "XML Document 1".

The variable doc2 here references the Document instance of the loaded "XML Document 2".

The DOM parser is namespace aware.

Assume no execution errors.

Options:

A- <ns:root2 xmlns:ns='urn:xmlmaster:EX2' ns:info='value' xmlns:ns='urn:xmlmaster:EX1' />

B- <ns:root2 NS1:info='' xmlns:ns='urn:xmlmaster:EX2' xmlns:NS1='urn:xmlmaster:EX1' />

C- <ns:root2 NS1:info='value' xmlns:ns='urn:xmlmaster:EX2' xmlns:NS1='urn:xmlmaster:EX1' />

D- <ns:root2 ns:info='value' xmlns:ns='urn:xmlmaster:EX2' />

Answer:

C

Question 11

Question Type: MultipleChoice

Which of the following DOM (Level 2) nodes is not a child node of an element node (Element) in DOM trees?

Options:

A- Element

B- Processing Instruction

C- Attr

D- Comment

Answer:

C

Question 12

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML Document," and process XML using "DOM Processing."

Options:

A- <Customer:root xmlns:ns='urn:xmlmaster:EX'
xmlns:Customer= ' urn:xmlmaster:EX ' >
<ns:data>string value</ns:data>
</Customer:root>

B- <Customer:root xmlns:ns='urn:xmlmaster:EX'
xmlns:Customer= ' urn:xmlmaster:EX ' >
<Customer:data>string value</Customer:data>
</Customer:root>

C- <Customer:root xmlns:Customer='urn:xmlmaster:EX'>
<Customer:data>string value</Customer:data>
</Customer:root>

D- <ns:root xmlns:ns='urn:xmlmaster:EX'>

<ns:data>string value</ns:data>
</ns:root>

Answer:

A