CertsDeals

Free Questions for I10-002 by certsdeals

Shared by Nelson on 09-08-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML Document". Choose the XML Schema Document that correctly defines the structure of "XML Document".

Options:

A-

```
<xs:schema
xmlns:xs= ' http://www.w3.org/2001/XMLSchema '
targetNamespace= ' urn:xmlmaster:testml '
xmlns:tns= ' urn:xmlmaster:testml ' >
<xs:element name= ' TestML ' type= ' tns:testmlType ' />
<xs:complexType name= ' testmlType ' >
<xs:sequence>
<xs:element ref= ' tns:record ' maxOccurs= ' unbounded ' />
</xs:sequence>
</xs:complexType>
<xs:element name= ' record ' type= ' tns:recordType ' />
<xs:complexType name= ' recordType ' >
< xs:attribute name='level' type= ' xs:int ' />
< xs:attribute name='data' type= ' xs:int ' />
</xs:complexType>
```

```
</xs:schema>
```

B-

```
<xs:schema
xmlns:xs= ' http://www.w3.org/2001/XMLSchema '
targetNamespace= ' urn:xmlmaster:testml '
xmlns:tns= ' urn:xmlmaster:testml ' >
<xs:element name= ' TestML ' type= ' tns:testmlType ' />
<xs:complexType name= ' tns:testmlType ' >
<xs:sequence>
<xs:element ref= ' tns:record ' maxOccurs= ' unbounded ' />
</xs:sequence>
</xs:complexType>
<xs:element name= ' record ' type= ' tns:recordType ' />
<xs:complexType name= ' tns:recordType ' >
< xs:attribute ref='tns:level' />
< xs:attribute ref='tns:data' />
</xs:complexType>
< xs:attribute name='tns:level' type= ' xs:int ' />
< xs:attribute name='tns:data' type= ' xs:int ' />
</xs:schema>
```

C-

```
<xs:schema
xmlns:xs= ' http://www.w3.org/2001/XMLSchema '
targetNamespace= ' urn:xmlmaster:testml '
xmlns:tns= ' urn:xmlmaster:testml ' >
<xs:element name= ' TestML ' type= ' tns:testmlType ' />
<xs:complexType name= ' testmlType ' >
```

```
<xs:sequence>
<xs:element ref= ' tns:record ' maxOccurs= ' unbounded ' />
</xs:sequence>
</xs:complexType>
<xs:element name= ' record ' type= ' tns:recordType ' />
<xs:complexType name= ' recordType ' >
< xs:attribute ref='tns:level' />
< xs:attribute ref='tns:data' />
</xs:complexType>
< xs:attribute name='level' type= ' xs:int ' />
< xs:attribute name='data' type= ' xs:int ' />
</xs:schema>
```

D- <xs:schema

```
xmlns:xs= ' http://www.w3.org/2001/XMLSchema '
targetNamespace= ' urn:xmlmaster:testml '
xmlns:tns= ' urn:xmlmaster:testml ' >
<xs:element name= ' TestML ' >
<xs:complexType>
<xs:sequence>
<xs:element name= ' record ' maxOccurs= ' unbounded ' >
<xs:complexType>
<xs:attribute name= ' tns:level ' type= ' xs:int ' />
<xs:attribute name= ' tns:data ' type= ' xs:int ' />
</xs:complexType>
</xs:element>
```

```
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

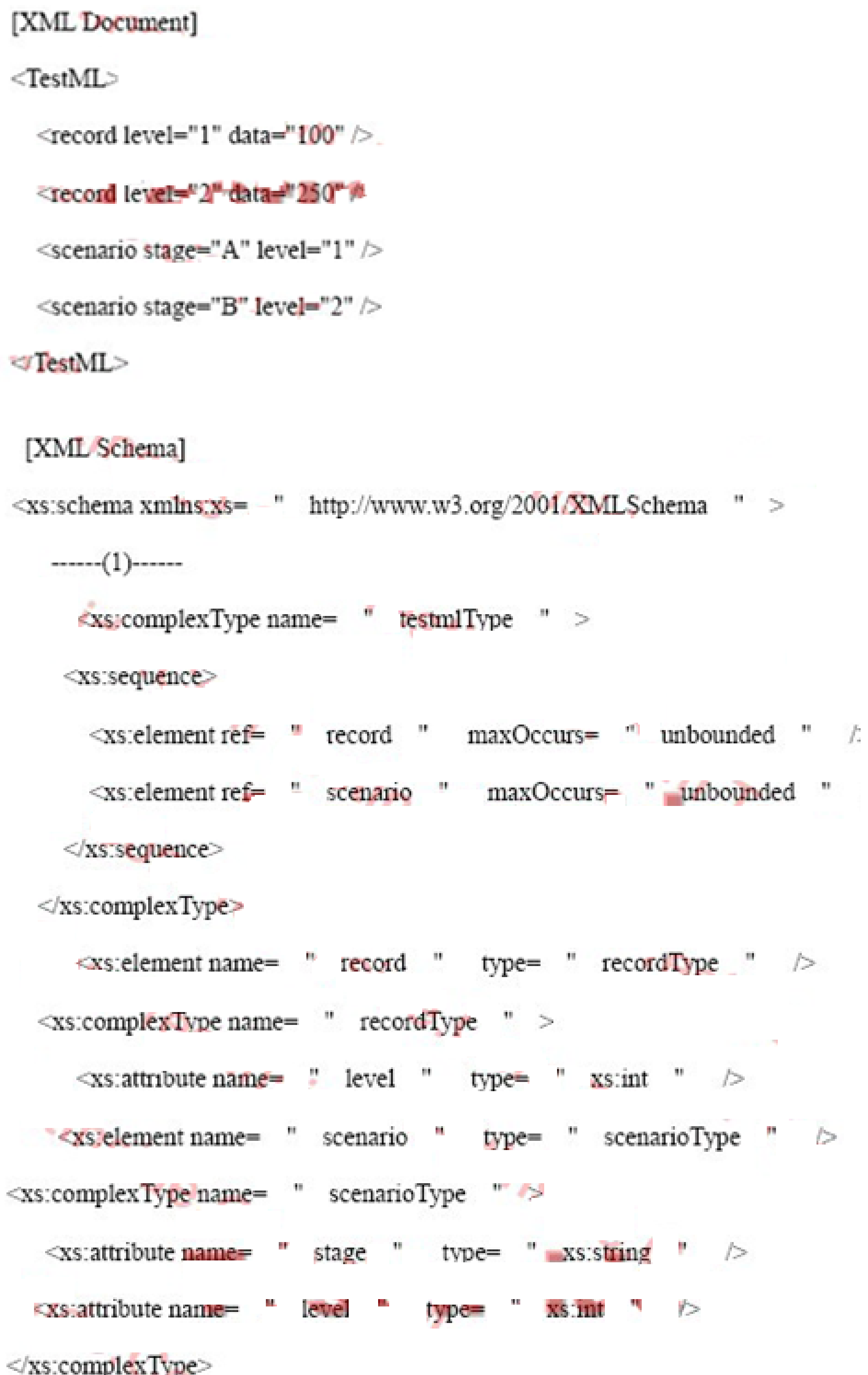
Answer:

C

Question 2

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 attribute of the record element must be singularly unique within the XML document, and further, that the level attribute of the scenario element must reference the value of the level attribute of the record element. Select which of the following correctly describes what should be written in " XML Schema " document (1).



```
[XML Document]
<TestML>
   <record level="1" data="100" />
   <record level="2" data="250" />
   <scenario stage="A" level="1" />
   <scenario stage="B" level="2" />
</TestML>

  [XML Schema]
<xs:schema xmlns:xs= " http://www.w3.org/2001/XMLSchema " >
     ------(1)------
         <xs:complexType name= " testmlType " >
       <xs:sequence>
          <xs:element ref= " record " maxOccurs= " unbounded " /:
          <xs:element ref= " scenario " maxOccurs= " unbounded "
       </xs:sequence>
    </xs:complexType>
         <xs:element name= " record " type= " recordType " />
    <xs:complexType name= " recordType " >
         <xs:attribute name= " level " type= " xs:int " />
       <xs:element name= " scenario " type= " scenarioType " />
<xs:complexType name= " scenarioType " >
    <xs:attribute name= " stage " type= " xs:string " />
   <xs:attribute name= " level " type= " xs:int " />
</xs:complexType>
```

Options:

```
A- <xs:element name='TestML' type='testmlType'>
< xs:unique name= ' LEVEL ' >
<xs:selector xpath= ' record ' />
<xs:field xpath= ' record/@level ' />
</ xs:unique >
< xs:ref name= ' levelRef ' refer= ' LEVEL ' >
<xs:selector xpath= ' scenario ' />
<xs:field xpath= ' scenario/@level ' />
</ xs:ref >
</xs:element>
```

```
B- <xs:element name='TestML' type='testmlType'>
< xs:unique name= ' LEVEL ' >
<xs:selector xpath= ' record ' />
<xs:field xpath= ' @level ' />
</ xs:unique >
< xs:ref name= ' levelRef ' refer= ' LEVEL ' >
<xs:selector xpath= ' scenario ' />
<xs:field xpath= ' @level ' />
</ xs:ref >
</xs:element>
```

```
C- <xs:element name='TestML' type='testmlType'>
```

```
< xs:key name= ' LEVEL ' >
<xs:selector xpath= ' record ' />
<xs:field xpath= ' record/@level ' />
</ xs:key >
< xs:keyref name= ' levelRef ' refer= ' LEVEL ' >
<xs:selector xpath= ' scenario ' />
<xs:field xpath= ' scenario/@level ' />
</ xs:keyref >
</xs:element>
```

D-

```
<xs:element name='TestML' type='testmlType'>
< xs:key name= ' LEVEL ' >
<xs:selector xpath= ' record ' />
<xs:field xpath= ' @level ' />
</ xs:key >
< xs:keyref name= ' levelRef ' refer= ' LEVEL ' >
<xs:selector xpath= ' scenario ' />
<xs:field xpath= ' @level ' />
</ xs:keyref >
</xs:element>
```

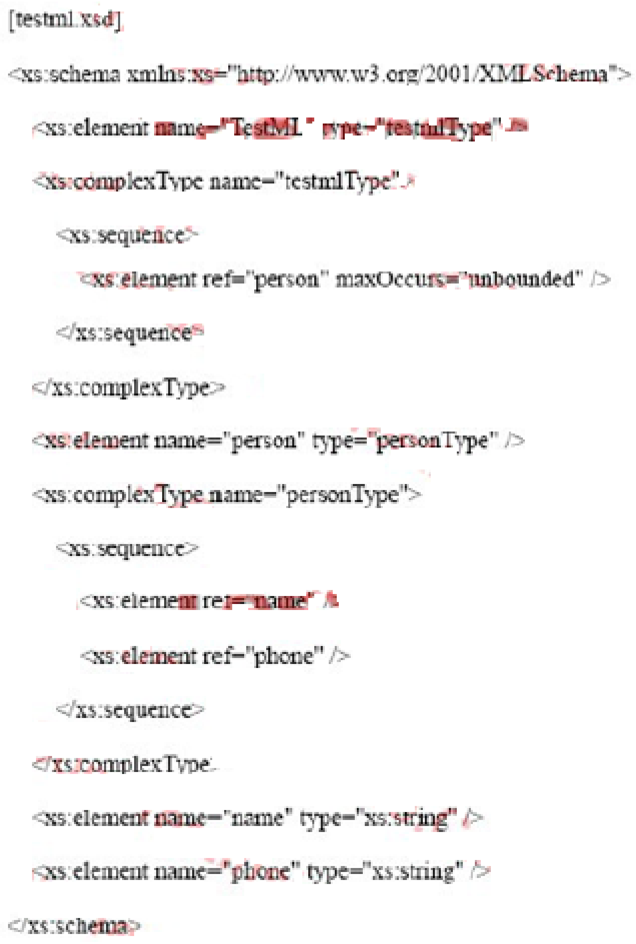
Answer:

D

Question 3

Question Type: **MultipleChoice**

Push the Exhibit Button to load the referenced "testml.xsd".



```
[testml.xsd]

<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="person" maxOccurs="unbounded" />

    </xs:sequence>

  </xs:complexType>

  <xs:element name="person" type="personType" />

  <xs:complexType name="personType">

    <xs:sequence>

      <xs:element ref="name" />

      <xs:element ref="phone" />

    </xs:sequence>

  </xs:complexType>

  <xs:element name="name" type="xs:string" />

  <xs:element name="phone" type="xs:string" />

</xs:schema>
```

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 to write a cell Phone element as a child element of the person 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 the XML parser correctly processes the XML schema Location attribute.

[XML.Documet]

John Smith

03-000-999

00-1111-2222

Options:

```
A- <xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:import schemaLocation='testml.xsd' />
<xs:complexType name='personType'>
<xs:sequence>
<xs:element ref= ' name ' />
<xs:element ref= ' phone ' />
<xs:element ref= ' cellPhone ' />
</xs:sequence>
</xs:complexType>
<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:complexType name='newPersonType' substitutionGroup='personType'>
```

```
<xs:sequence>
<xs:element ref= ' name ' />
<xs:element ref= ' phone ' />
<xs:element ref= ' cellPhone ' />
</xs:sequence>
</xs:complexType>
<xs:element name= ' cellPhone ' type= ' xs:string ' />
</xs:schema>
```

C-

```
<xs:schema xmlns:xs='http://www.w3.org/2001/XMLSchema'>
<xs:redefine schemaLocation= ' testml.xsd ' >
<xs:complexType name= ' personType ' >
<xs:complexContent>
<xs:extension base= ' personType ' >
<xs:sequence>
<xs:element ref= ' cellPhone ' />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
</xs:redefine>
<xs:element name= ' cellPhone ' type= ' xs:string ' />
</xs:schema>
```

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

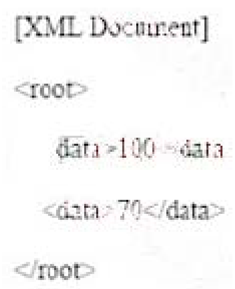
Answer:

C

Question 4

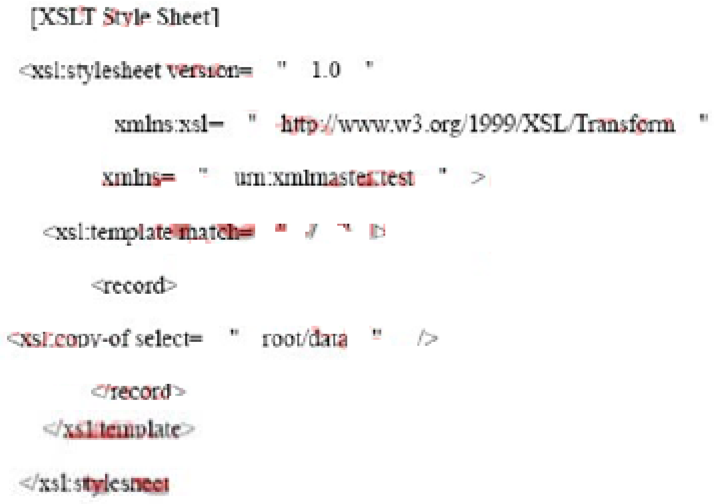
Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML Document".



```
[XML Document]
<root>
   <data>100</data>
   <data>70</data>
</root>
```

Select which of the following correctly describes the output results of an XSLT transformation of the "XML Document" using the "XSLT Style Sheet". 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 "
       xmlns=" urn:xmlmaster:test " >
  <xsl:template match=" / " >
      <record>
<xsl:copy-of select=" root/data " />
      </record>
  </xsl:template>
</xsl:stylesheet>
```

Options:

A- <record>

<data>100</data>

</record>

B- <record xmlns='urn:xmlmaster:test'>
<data>100</data>
</record>

C- <record xmlns='urn:xmlmaster:test'>
<data xmlns= '' >100</data>
</record>

D- <record>
<data>100</data>
<data>70</data>
</record>

E- <record xmlns='urn:xmlmaster:test'>
<data>100</data>
<data>70</data>
</record>

F- <record xmlns='urn:xmlmaster:test'>
<data xmlns= '' >100</data>
<data xmlns= '' >70</data>
</record>

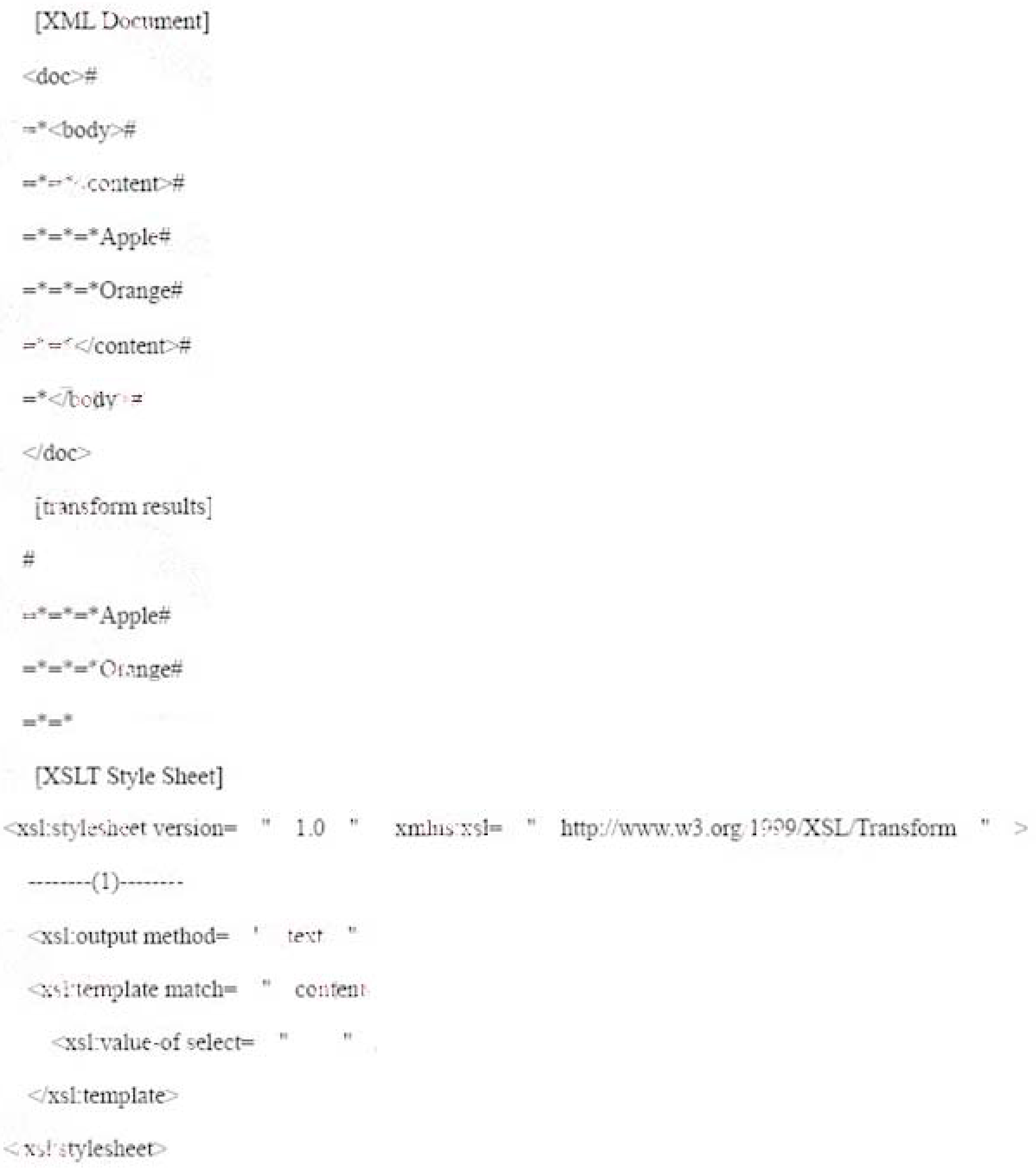
Answer:

F

Question 5

Question Type: MultipleChoice

What must you write in XSLT style sheet (1) to process the following "XML Document" and obtain the following "transform results"? Select the correct answer below. Note that "#" indicates a line feed, and "=*" indicates a tab. Assume that the XSLT processor can output transformation results as a document.



```
  [XML Document]
 <doc>#
 =*<body>#
 =*=*<content>#
 =*=*=*Apple#
 =*=*=*Orange#
 =*=*</content>#
 =*</body>#
 </doc>
  [transform results]
 #
 =*=*=*Apple#
 =*=*=*Orange#
 =*=*
  [XSLT Style Sheet]
<xsl:stylesheet version=  "  1.0  "   xmlns:xsl=  "  http://www.w3.org/1999/XSL/Transform  "  >
  --------(1)--------
  <xsl:output method=  '  text  "
  <xsl:template match=  "  content
    <xsl:value-of select=  "      "
  </xsl:template>
<xsl:stylesheet>
```

Options:

A- Nothing needs to be written.

B- <xml:space='preserve'/>

C- <xsl:preserve-space elements='content'/>

D- <xsl:strip-space elements='doc body'/>

Answer:

D

Question 6

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML Document".

Assume that the character "3" is obtained from the "XML document". Select which XSLT style sheet correctly performs the transformation. (Multiple answers possible. Select two.)

Options:

A-
```
<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' //data[x='1'][y='2'] ' />
</xsl:template>
</xsl:stylesheet>
```

B-
```
<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' //data[(attribute::x='1') and (text()='3')] ' />
</xsl:template>
</xsl:stylesheet>
```

C-
```
<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' //data[self='3'] ' />
</xsl:template>
</xsl:stylesheet>
```

D-
```
<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' //data[self::*='3'] ' />
</xsl:template>
</xsl:stylesheet>
```

Answer:

B, D

Question 7

Question Type: **MultipleChoice**

Push the Exhibit Button to load the referenced "XML Document".

[XML Document]

lmnop

Assume that the "XML document" is changed to the "Results XML Document." Select which XSLT style sheet correctly performs the transformation.

Note that the XSLT processor can output transformation results as a document.

[Results XML Document]

Or

Options:

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

```
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' root/data ' />
</xsl:template>
<xsl:template match= ' data ' >
<xsl:element name='<xsl:value-of select='.'/>'/>
</xsl:template>
</xsl:stylesheet>
```

```
B- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' root/data ' />
</xsl:template>
<xsl:template match= ' data ' >
<xsl:element name='{ . }'/>
</xsl:template>
</xsl:stylesheet>
```

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

```
D- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match= ' / ' >
```

```
<xsl:apply-templates select= ' root/data ' />
</xsl:template>
<xsl:template match= ' data ' >
<xsl:text disable-output-escaping='no'><</xsl:text>
<xsl:value-of select='.'/>
<xsl:text disable-output-escaping='no'>/></xsl:text>
</xsl:template>
</xsl:stylesheet>
```

Answer:

B

Question 8

Question Type: MultipleChoice

Push the Exhibit Button to load the referenced "XML document".

[XML Document]

lmnop

Assume that the "XML Document" is changed to the "Results XML Document." Select which XSLT style sheet correctly performs the transformation. Note that the XSLT processor can output transformation results as a document.

[Results XML Document]

lmnop

Options:

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

```
<xsl:include href='exam.xsl' />
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' root ' />
</xsl:template>
<xsl:template match= ' root ' >
<AAA><BBB><xsl:value-of select= ' data ' /></BBB></AAA>
</xsl:template>
</xsl:stylesheet>
[exam.xsl]
<xsl:stylesheet version= ' 1.0 ' xmlns:xsl= ' http://www.w3.org/1999/XSL/Transform ' >
<xsl:template match= ' //root ' >
<ZZZ><YYY><xsl:value-of select= ' data ' /></YYY></ZZZ>
</xsl:template>
</xsl:stylesheet>
```

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

```
<xsl:import href='exam.xsl' />
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' root ' />
</xsl:template>
<xsl:template match= ' root ' >
<AAA><BBB><xsl:value-of select= ' data ' /></BBB></AAA>
</xsl:template>
</xsl:stylesheet>
[exam.xsl]
<xsl:stylesheet version= ' 1.0 ' xmlns:xsl= ' http://www.w3.org/1999/XSL/Transform ' >
<xsl:template match= ' //root ' >
<ZZZ><YYY><xsl:value-of select= ' data ' /></YYY></ZZZ>
</xsl:template>
</xsl:stylesheet>
```

```
C- <xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:include href='exam.xsl' />
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' root ' />
</xsl:template>
<xsl:template match= ' root ' >
<AAA><BBB><xsl:value-of select= ' data ' /></BBB></AAA>
</xsl:template>
</xsl:stylesheet>
[exam.xsl]
<xsl:stylesheet version= ' 1.0 ' xmlns:xsl= ' http://www.w3.org/1999/XSL/Transform ' >
```

```
<xsl:template match= ' root ' >
<ZZZ><YYY><xsl:value-of select= ' data ' /></YYY></ZZZ>
</xsl:template>
</xsl:stylesheet>
```

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

```
<xsl:import href='exam.xsl' />
<xsl:template match= ' / ' >
<xsl:apply-templates select= ' root ' />
</xsl:template>
<xsl:template match= ' root ' >
<AAA><BBB><xsl:value-of select= ' data ' /></BBB></AAA>
</xsl:template>
</xsl:stylesheet>
[exam.xsl]
<xsl:stylesheet version= ' 1.0 ' xmlns:xsl= ' http://www.w3.org/1999/XSL/Transform ' >
<xsl:template match= ' root ' >
<ZZZ><YYY><xsl:value-of select= ' data ' /></YYY></ZZZ>
</xsl:template>
</xsl:stylesheet>
```

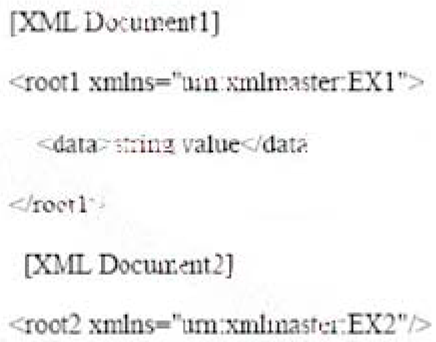
Answer:

A

Question 9

Question Type: MultipleChoice

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



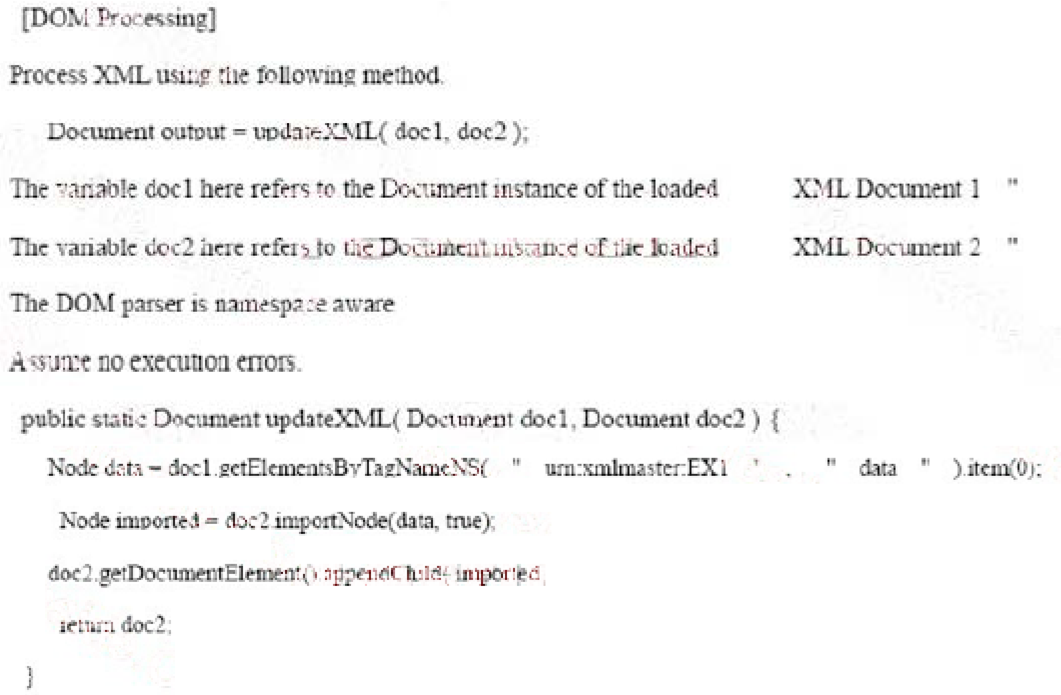
[XML Document1]

```
<root1 xmlns="urn:xmlmaster:EX1">
  <data>string value</data>
</root1>
```

[XML Document2]

```
<root2 xmlns="urn:xmlmaster:EX2"/>
```

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 refers to the Document instance of the loaded XML Document 1 "

The variable doc2 here refers to the Document instance of the loaded XML Document 2 "

The DOM parser is namespace aware

Assume no execution errors.

```
public static Document updateXML( Document doc1, Document doc2 ) {
   Node data = doc1.getElementsByTagNameNS( " urn:xmlmaster:EX1 ", " data " ).item(0);
    Node imported = doc2.importNode(data, true);
   doc2.getDocumentElement().appendChild( imported
    return doc2;
 }
```

Options:

A- <root2 xmlns='urn:xmlmaster:EX2'>
<data xmlns= ' urn:xmlmaster:EX1 ' >string value</data>
</root2>

B- <root2 xmlns='urn:xmlmaster:EX2'>
<data>string value</data>
</root2>

C- <root2 xmlns='urn:xmlmaster:EX2'>
<data xmlns= ' urn:xmlmaster:EX1 ' />
</root2>

D- <root2 xmlns='urn:xmlmaster:EX2'>
<data/>
</root2>

Answer:

A

Question 10

Question Type: **MultipleChoice**

Which of the following correctly describes the DOM (Level 2) Node interface?

Options:

A- The Node interface can be used to change the value (nodeValue) of the DOM element node (Element)

B- The Node interface can be used to change the name (nodeName) of the DOM element node (Element)

C- The Node interface can be used to change the value (nodeValue) of the DOM attribute node (Attr)

D- The Node interface can be used to change the name (nodeName) of the DOM attribute node (Attr)

Answer:

C