



Exam : XML Master Certification
I10-002

Title : XML Master: Professional V2

Version : R6.1

Prepking - King of Computer Certification

Important Information, Please Read Carefully

Other Prepking products

A) Offline Testing engine

Use the offline Testing engine product to practice the questions in an exam environment.

B) Study Guide (not available for all exams)

Build a foundation of knowledge which will be useful also after passing the exam.

Latest Version

We are constantly reviewing our products. New material is added and old material is updated. Free updates are available for 90 days after the purchase. You should check your member zone at Prepking and update 3-4 days before the scheduled exam date.

Here is the procedure to get the latest version:

1. Go to www.Prepking.com

2. Click on **Member zone/Log in (right side)**

3. Then click My Account

4. The latest versions of all purchased products are downloadable from here. Just click the links.

For most updates, it is enough just to print the new questions at the end of the new version, not the whole document.

Feedback

If you spot a possible improvement then please let us know. We are always interested in improving product quality.

Feedback should be sent to feedback@Prepking.com. You should include the following:

Exam number, version, page number, question number, and your login ID.

Our experts will answer your mail promptly.

Copyright

Each PDF file contains a unique serial number associated with your particular name and contact information for security purposes. So if we find out that a particular PDF file is being distributed by you, Prepking reserves the right to take legal action against you according to the International Copyright Laws.

Explanations

This product does not include explanations at the moment. If you are interested in providing explanations for this exam, please contact feedback@Prepking.com.

1. Select which of the following correctly describes WSDL. (WSDL 1.1)

A. WSDL assumes SOAP as the message transmission form

B. When WSDL is defined by a combination of style="rpc" and use="encoded", then encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" must be designated

C. When WSDL is defined by a combination of style="rpc" and use="encoded", then the encodingStyle attribute cannot be designated

D. WSDL may be defined by a combination of style="rpc" and use="literal"

Answer: D

2. Which of the following describes the most correct call order of the ContentHandler interface methods when parsing the following "XML Document" using a non-validating SAX parser? This question reflects line feeds within the XML document.

[XML Document]

<a>

c

A. startDocument - startElement - characters - startElement - characters - characters - characters - endElement - characters - endElement - endDocument

B. startDocument - startElement - ignorableWhitespace - startElement - ignorableWhitespace - characters - ignorableWhitespace - endElement - ignorableWhitespace - endElement - endDocument

C. startDocument - startElement - startElement - characters - endElement - endElement - endDocument

D. startDocument - startElement - startElement - characters - characters - endElement - endElement - endDocument

Answer: A

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

[XML Document]

<TestML xmlns="urn:xmllmaster:testml">

<record level="1" data="100" />

<record level="2" data="250" />

</TestML>

Choose the XML Schema Document that does not correctly define the structure of the "XML Document".

A. <xs:schema

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

targetNamespace="urn:xmllmaster: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="urn:xmlmaster:testml" >
<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:complexType name="recordType">
    <xs:attribute name="level" type="xs:int" />
    <xs:attribute name="data" type="xs:int" />
  </xs:complexType>
</xs:schema>

```

C. <xs:schema

```

        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="urn:xmlmaster:testml" >
<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:complexType name="recordType">
    <xs:attribute name="level" type="xs:int" />
    <xs:attribute name="data" type="xs:int" />
</xs:complexType>
</xs:schema>

```

D. <schema

```

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

```

Answer: C

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

- 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

5. 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;  
}
```

- 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

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

[XML Document]

```
<root><data>lmnop</data></root>
```

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]

```
<ZZZ><YYY>lmnop</YYY></ZZZ>
```

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

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

[XML Document]

```
<root><data>Imnop</data></root>
```

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]

```
<Imnop/>
```

Or

```
<Imnop></Imnop>
```

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

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

[XML Document]

<root>

<data x="1" y="2">3</data>

<data x="1" y="4">6</data>

<data x="1" y="6">9</data>

<data x="1" y="8">12</data>

</root>

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.)

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= " / " >

100% Pass Guaranteed or Full Refund

Word to Word Real Exam Questions from Real Test

Buy full version of exam from this link below

<http://www.prepking.com/I10-002.htm>