

I10-002 Practice Test

XML Master XML Master: Professional V2

I10-002: XML Master: Professional V2

Practice Exam: I10-002 Exams

Exam Number/Code: I10-002

Exam Name: XML Master: Professional V2

Questions and Answers: 80 Q&As

(XML Master)



Exam : [I10-002](#)

For candidates making preparation for the XML Master I10-002 questions and answers, what they most desire is to easily pass the I10-002 (XML Master:

Professional V2) exam. ItCertHome I10-002 includes 80 questions and answers, which are collected and collated by experts of XML Master. With our I10-002 study materials, you can successfully take XML Master certification of I10-002 exam and go further on XML Master career path.

Free I10-002 Demo

we provide XML Master I10-002 demo exam for free (in PDF format) before you decide to purchase it. Thus,you can know better about the quality of our practice exam and then make your right decision.

ItCertHome Test tool's advantages:

High Quality and Value of I10-002 exam

ItCertHome XML Master XML Master I10-002 Practice Questions is developed and finished by XML Master technical team , cover every field of the exam . Besides, we have verified I10-002 answers,almost 100% correct.

100% Guarantee to Pass I10-002 Exam

We promise to give you FULL REFUND if you fail the XML Master I10-002 certification(XML Master Certified Network Associate) with the use of our ItCertHome testing engine.

I10-002 Self Test Mode

ItCertHome provides a simulated and interactive environment where you can test your knowledge and skills about I10-002 to ensure enough assurance in real testing center.

Periodic Updates of ItCertHome I10-002

Once there is some change on XML Master I10-002 exam, we will update it timely, and the product you buy will be updated within 90 days for free.

Professional and Efficient Service

We offer 7*24 customer support via diverse channels: LIVE CHAT,MAIL.Expecting the communication with you about IT certification.

The Questions & Answers cover the latest real test and with all the correct answer.we promise the Q&A for **XML Master XML Master I10-002** examination of original title complete coverage.ItCertHome **I10-002 Braindumps** Questions & Answers help you pass the exam. Otherwise,we will give you a full refund.

We promised that use ItCertHome Q&A ensure you pass the exam at your first try.

1. The Q&A are from cooperation exam center of the real original title,summaried by our professional team and collated by senior IT lectured in co-training center to make sure the professional quality of the Q&A.The correctly of the Q&A is 100%, the coverage of I10-002 Q&A are more than 96%.All you need to study the whole I10-002 Q&A before you participate the certification exam,it should be possible to easily complete the exam and pass the certification.

2. ItCertHome to all the Q&A, we promised "do not pass the exam give you a full refund". If you buy our I10-002 Q&A and did not pass the exam at the first try. You can take the examination report card that stamped with PROMETRIC or VUE test centers Seal. we will refund your full cost of I10-002 Q&A, absolutely guarantee you interests have no losses.(For a full refund details)

3. ItCertHome professional IT Q&A vendors, we provide well after-sale service. To all the customers buy the Q&A, we provide track service. when you buy the Q&A with in one year. you can enjoy the upgrade Q&A service for free. If in this period, the certified test center change the I10-002 Q&A, we will update the Q&A in the first time, and provide you the download update for free.

[XML Master I10-002](#) Test belongs to one of the XML Master certified test, if needs to obtain the XML Master certificate, you also need to participate in other related test, the details you may visit the XML Master certified topic, in there, you will see all related XML Master certified subject of examination.

I10-002

ItCertHome professional provide XML Master I10-002 the newest Q&A, completely covers I10-002 test original topic. With our complete XML Master resources, you will minimize your XML Master cost and be ready to pass your I10-002 tests on Your First Try, 100% Money Back Guarantee included!

This I10-002 PDF demo do not include the questions and answers's picture:

Exam : XML Master Certification

I10-002

Title : XML Master: Professional V2

1. 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.

- 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

2. 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]

```
<ZZZ><YYY>Imnop</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.xml]

```
<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.xml" />
<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.xml]

```
<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.xml" />
<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.xml]

```
<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.xml" />
<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.xml]

```
<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

3. 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).

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

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

[XML Document]

```
<TestML xmlns="urn:xmlmaster: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: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="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

5. 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 to write a cellPhone 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 schemaLocation attribute.

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

6. Select which of the following DOM (Level 2) nodes does not hold a value (returns "null"). (Multiple answers possible. Select two.)

A. Attr

B. Comment

C. Element

D. Document

Answer: CD

7. 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.)

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: BD

8. 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.

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

9. Select which of the following correctly describes WSDL. (WSDL 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

10. 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".

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

- 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

11. 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

12. Select which of the following correctly describes the results of performing a validation check on "XML Document". Assume that the XML parser correctly processes the XML schema noNamespaceSchemaLocation attribute and the schemaLocation attribute.

- A. Valid
- B. The coding for the XML Schema Document is not appropriate; therefore, an error is thrown (initial error) when processing the "testml.xsd" import element
- C. The coding for the XML Schema Document is not appropriate; therefore, an error is thrown (initial error) when processing the "testml.xsd" "<xs:element ref="rec:record" maxOccurs="unbounded" />"
- D. No processing error, but is not valid.

Answer: A

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

```

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

14. 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>
<b>
c
</b>
</a>
```

- 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 - endDocumentW

Answer: A

15. 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

More [I10-002 practice test](#)

Related I10-002 Exams

[I10-001](#) XML Master Basic V2

[I10-002](#) XML Master: Professional V2

[I10-003](#) XML Master Professional Database Administrator

Other XML Master Exams

[I10-001](#)

[I10-002](#)

[I10-003](#)