



# I10-002<sup>Q&As</sup>

XML Master: Professional V2

**Pass XML Master I10-002 Exam with 100% Guarantee**

Free Download Real Questions & Answers **PDF** and **VCE** file from:

<https://www.pass4itsure.com/i10-002.html>

100% Passing Guarantee  
100% Money Back Assurance

Following Questions and Answers are all new published by XML Master  
Official Exam Center

-  **Instant Download** After Purchase
-  **100% Money Back** Guarantee
-  **365 Days** Free Update
-  **800,000+** Satisfied Customers



**QUESTION 1**

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

```
[XML Document]
<Unfinished xmlns="urn:xlmaster:UNFINISHED">
  <Item>
    <Worker xmlns="urn:xlmaster:WORKER"
Name="Jeff Workman"/>
    <Activity xmlns="urn:xlmaster:VISIT"
Code="0002">
      <DateTime>2004-07-01 13:00</DateTime>
      <Job Procedure="Install air conditioner"
Time="2"/>
      <Status>WAITING</Status>
    </Activity>
  </Item>
  <Item>
    <Worker xmlns="urn:xlmaster:WORKER" Name="Jim
Worker"/>
    <Activity xmlns="urn:xlmaster:VISIT"
Code="0003">
      <DateTime>2004-07-01 16:00</DateTime>
      <Job Procedure="Clean air conditioner"
Time="2"/>
      <Status>WAITING</Status>
    </Activity>
  </Item>
</Unfinished>
```

When processing the "XML Document" according to the method shown by "DOM Processing," 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. Assume that the processed XML Document has no indents (ignorable white space such as line feeds, tabs, etc.).



[DOM Processing]

Create XML using the following method.

```
Document output = createXML( doc, impl );
```

The variable doc here references the Document instance of the loaded XML Document.

The variable impl here references the DOMImplementation instance.

The DOM parser is namespace aware.

Assume no execution errors.

```
public static Document createXML( Document doc, DOMImplementation impl ) {
    String name = "Jim Worker";

    Document output = impl.createDocument(
        "urn:xmlmaster:LIST", "List", null);
    Element root = output.getDocumentElement();

    NodeList nl = doc.getElementsByTagNameNS("urn:xmlmaster:WORKER", "Worker");
    for (int i = 0; i < nl.getLength(); i++) {
        Element e1 = (Element)nl.item(i);
        if (e1.getAttribute("Name").equals(name)) {
            Element elem = output.createElementNS(
                "urn:xmlmaster:LIST", "VisitList");
            root.appendChild(output.importNode(e1, true));
            Element e2 = (Element)e1.getNextSibling();
            root.appendChild(output.importNode(e2, true));
        }
    }
    return output;
}
```

- A. 2004-07-01 16:00 WAITING
- B. 2004-07-01 16:00 WAITING
- C. C. 2004-07-01 16:00 WAITING
- D. D.D. 2004-07-01 16:00 WAITING

Correct Answer: C

## QUESTION 2

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

```
[XML Document]
<?xml version="1.0"?>
<!DOCTYPE a [
<!ELEMENT a {#PCDATA | b} *>
<!ELEMENT b {#PCDATA}>
]>
<a>
<b>c</b>
</a>
```

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



-endDocument

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

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

Correct Answer: B

### QUESTION 3

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 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.



[XML Document]

```
<TestML>
  <person>
    <name>John Smith</name>
    <phone>03-0000-9999</phone>
    <cellPhone>000-1111-2222</cellPhone>
  </person>
</TestML>
```

- 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.

A. Option A

B. Option B

C. Option C

D. Option D

Correct Answer: C

**QUESTION 4**

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

```
[XML Document]
<Unfinished xmlns='urn:xmlmaster:UNFINISHED">
  <List>
    <Worker xmlns='urn:xmlmaster:WORKER"
Name="Jeff Workman"/>
    <Activity xmlns="urn:xmlmaster:VISIT"
Code="0002">
      <DateTime>2004-07-01 13:00</DateTime>
      <Location xmlns="urn:xmlmaster:CONTACT">
        <Name>John Smith</Name>
        <Address>Shinjuku Ward Tokyo</Address>
      </Location>
      <Job Procedure="Install Air Conditioner"
Time="2"/>
      <Status>COMPLETED</Status>
    </Activity>
    <Activity xmlns="urn:xmlmaster:VISIT"
Code="0003">
      <DateTime>2004-07-01 16:00</DateTime>
      <Location xmlns="urn:xmlmaster:CONTACT">
        <Name>Ken Miller</Name>
        <Address>Shibuya Ward Tokyo</Address>
      </Location>
      <Job Procedure="Clean air conditioner"
Time="2"/>
      <Status>WAITING</Status>
    </Activity>
  </List>
</Unfinished>
```

When processing the "XML Document" according to the method shown by "DOM Processing," which of the

following correctly describes the output results (print and/or println method output)? Although the expected processing result is choice "A", processing may not occur as expected.



```
[DOM Processing]
Process XML using the following method.
    printXML( doc );
The variable doc here references the Document instance of the loaded XML Document.
The DOM parser is namespace aware.
Assume no execution errors.

public static void printXML( Document doc ) {
    Node node = doc.getElementsByTagNameNS(
        "urn:xmlmaster:CONTACT", "Name").item(0);
    String name = node.getFirstChild().getNodeValue();

    NodeList nl = doc.getElementsByTagNameNS("urn:xmlmaster:VISIT", "Activity");
    for (int i = 0; i < nl.getLength(); i++) {
        Element elem = (Element)nl.item(i);
        node = elem.getElementsByTagNameNS(
            "urn:xmlmaster:VISIT", "DateTime").item(0);
        System.out.print("Not Visited:");
        System.out.print(node.getFirstChild().getNodeValue());
        System.out.print(" Worker=" + name);
        node = elem.getElementsByTagNameNS(
            "urn:xmlmaster:CONTACT", "Name").item(0);
        System.out.print(", Visit (");
        System.out.print(node.getFirstChild().getNodeValue());
        System.out.println(")");
    }
}
```

- A. Not Visited:2004-07-01 13:00 Worker=Jeff Workman, Visit(John Smith) Not Visited:2004-07-01 16:00 Worker=Jeff Workman, Visit(Ken Miller)
- B. Not Visited:2004-07-01 13:00 Worker=John Smith, Visit(John Smith) Not Visited:2004-07-01 16:00 Worker=John Smith, Visit(Ken Miller)
- C. Not Visited:2004-07-01 13:00 Worker=Jeff Workman, Visit(John Smith)
- D. Not Visited:2004-07-01 13:00 Worker=John Smith, Visit(John Smith)

Correct Answer: B

## QUESTION 5

Which of the following is incorrect with respect to general characteristics of DOM (Level 2) processing and data binding tool processing?

- A. Data types such as integers and dates cannot be handled by DOM; however, a data binding tool can be used to handle the data type
- B. A validation against the schema cannot be performed with the DOM; however, a data binding tool can be used to perform a validation against the schema
- C. A DOM parser cannot be used to execute a program created to use a different DOM parser; however, a data binding tool can be used to execute a program created to use a different data binding tool
- D. For a compiled programming language, compared to DOM processing, a data binding tool can detect more element name mistakes and other programming errors during compiling

Correct Answer: C



VCE & PDF

Pass4itSure.com

<https://www.pass4itsure.com/i10-002.html>

2024 Latest pass4itsure I10-002 PDF and VCE dumps Download

---

[Latest I10-002 Dumps](#)

[I10-002 VCE Dumps](#)

[I10-002 Study Guide](#)