

XML I10-002

XML Master: Professional V2

Version: 4.0

QUESTION NO: 1

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

Explanation:

QUESTION NO: 2

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

- A. appendChild
- B. hasAttribute
- C. setAttribute
- D. setAttributeNode

Answer: C,D

Explanation:

QUESTION NO: 3

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

Answer: B

Explanation:

QUESTION NO: 4

Under the SAX specification, which of the following is a method for determining whether to validate a document with the SAX parser?

- A. Use the getFeature method of XMLReader interface to look up the feature value under the name "http://xml.org/sax/features/validation"
- B. Use the parse method of XMLReader interface to investigate whether an error occurs
- C. Use the parseWithValidating method of XMLReader interface
- D. Use the isValidatingParser method of XMLReader interface

Answer: A

Explanation:

QUESTION NO: 5

When processing the following "XML Document" according to the method shown by "SAX Processing," which of the following correctly describes the output results (print method output)?

```
[XML Document]
<doc><content>East<![CDATA[&]]>West</content></doc>
```

[SAX Processing]

Use the following "ContentHandlerImpl" class, and parse the XML Document using SAX. Assume no execution errors.

```
class ContentHandlerImpl extends DefaultHandler {
    public void characters(char[] ch, int start, int length) throws SAXException {
        System.out.print(new String(ch, start, length));
    }
}
```

- A. East<![CDATA[&]]>West
- B. East&West
- C. EastWest
- D. East

Answer: B

Explanation:

QUESTION NO: 6

Process the following "XML Document" according to the method shown by "DOM Processing," and output "article" (by print method). Select which of the following correctly describes what should be in (1) for "DOM Processing"

```
[XML Document]
<content>article</content>
```

[DOM Processing]

Process XML using the following method.

```
printXML( doc );
The variable doc here references the Document instance of the loaded XML document.
```

```
public static void printXML( Document doc ) {
    Element content = doc.getDocumentElement();
    System.out.print( -----(1)----- );
}
```

- A. content.getText()
- B. content.getNodeValue()
- C. content.getChildNode().getNodeValue()
- D. content.getFirstChild().getNodeValue()

Answer: D

Explanation:

QUESTION NO: 7

Push the Exhibit Button to load the referenced "XML Document," 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( 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 Document updateXML( Document doc ) {
  Node data1 = doc.getElementsByTagNameNS( "urn:xmlmaster:EX", "data1" ).item(0);
  Node data2 = doc.getElementsByTagNameNS( "urn:xmlmaster:EX", "data2" ).item(0);

  data2.appendChild( data1 );

  return doc;
}
```

- A.** <root xmlns="urn:xmlmaster:EX">
 <data1>string value</data1>
 <ns:data2 xmlns:ns="urn:xmlmaster:EX">
 <data1>string value</data1>
 </ns:data2>
 </root>
- B.** <root xmlns="urn:xmlmaster:EX">
 <data1>string value</data1>
 <data2>
 <data1>string value</data1>
 </data2>
 </root>
- C.** <root xmlns="urn:xmlmaster:EX">
 <ns:data2 xmlns:ns="urn:xmlmaster:EX" xmlns="">
 <data1 xmlns="urn:xmlmaster:EX">string value</data1>
 </ns:data2>
 </root>
- D.** <root xmlns="urn:xmlmaster:EX">
 <data2>
 <data1>string value</data1>
 </data2>
 </root>

Answer: C

Explanation:

QUESTION NO: 8

When processing the following "XML Document" according to the method shown by "DOM Processing," which of the following correctly describes the output results (print method output)?

```
[XML Document]
<doc><content>Today's<b>hot</b>news</content></doc>

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

public static void printXML( Document doc ) {
    Element content = (Element)doc.getDocumentElement().getFirstChild();
    NodeList nl = content.getChildNodes();
    System.out.print( nl.getLength() );

    Element bold = (Element)content.getElementsByTagName( "b" ).item(0);
    content.removeChild( bold );
    System.out.print( nl.getLength() );
}
```

- A. 44
- B. 43
- C. 42
- D. 33
- E. 32
- F. 31

Answer: E

Explanation:

QUESTION NO: 9

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

```
[XML Document1]
<ns:root1 xmlns:ns="urn:xmlmaster:EX1">
  <ns:data ns:info="value">string value</ns:data>
</ns:root1>

[XML Document2]
<ns:root2 xmlns:ns="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 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.

```
public static Document updateXML( Document doc1, Document doc2 ) {
    String NS = "urn:xmlmaster:EX1";
    Element data = (Element)doc1.getElementsByTagNameNS( NS, "data" ).item(0);
    Attr info = data.getAttributeNodeNS( NS, "info" );

    Attr imported = (Attr)doc2.importNode( info, true );
    doc2.getDocumentElement().setAttributeNodeNS( imported );

    return doc2;
}
```

- 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

Explanation:

QUESTION NO: 10

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>