**Q1. You are working with this XML code snippet from the XML document cars.xml. You need to return the information about the cars built after the year 2000. What does your XQuery look like?**

```
<cars>
    <car><make>Cadillac</make><model>Escalade</model><year>2007</year></car>
    <car><make>Cadillac</make><model>Escalade</model><year>2011</year></car>
    <car><make>Ford</make><model>Mustang</model><year>1968</year></car>
    <car><make>Ford</make><model>Mustang</model><year>1998</year></car>
    <car><make>Mercedes</make><model>C-Class</model><year>1999</year></car>
    <car><make>Mercedes</make><model>C-Class</model><year>2009</year></car>
</cars>
```

- [x] `doc("cars.xml")/cars/car[year>2000].data`
- [ ] `doc("cars.xml")/cars/car[xs:integer(year) gt 2000]`
- [ ] `doc("cars.xml")/cars/car[year gt 2000]`
- [ ] `doc("cars.xml")/cars/car[integer(year) > 2000]`

**Q2. You are working with the following XSD fragment. What does it say about the `<car>` element?**

```
<xs:element name="car">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="make" type="xs:string"/>
            <xs:element name="model" type="xs:string"/>
            <xs:element name="year" type="xs:string"/>
        </xs:sequence>
        <xs:anyAttribute/>
    </xs:complexType>
</xs:element>
```

- [ ] The `<car>` element can be extended with only one attribute
- [x] The `<car>` element can be extended with multiple attributes
- [ ] The `<car>` element cannot have any attributes
- [ ] The `<car>` element has child elements which can appear in any order

**Reference**: XSD The `<anyAttribute>` Element

**Q3. You are converting your HTML file into XHTML Strict. Which code snippet will validate without errors?**

- [ ]

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head><title>XHTML Example</title></head>
    <body bgcolor="#FFFFFF" >
        <p>Content goes here ...</p>
    </body>
</html>
```

- [ ]
```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head><title>XHTML Example</title></head>
    <body name="bodySection">
        <p><b>Content goes here ...</b></p>
    </body>
</html>
```

- [ ]
```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head><title>XHTML Example</title></head>
    <body color="#333333">
        <p><i>Content goes here ...</i></p>
    </body>
</html>
```

- [x]
```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head><title>XHTML Example</title></head>
    <body id="bodySelection">
        <p><strong>Content goes here ...</strong></p>
    </body>
</html>
```

**Q4. When working with Ajax applications, which is faster, XML or JSON?**

- [ ] XML, because it is extensible
- [x] JSON, because it transfers data without waiting for a server response
- [ ] XML, because it supports namespaces
- [ ] JSON, because it is already parsed into a JavaScript object

**Q5. Asynchronous Javascript and XML (Ajax) is technique for creating better, faster, and more interactive web applications. In addition to JavaScript and XML on the back end, which technologies are commonly used to craft AJAX experiences on the front end?**

- [ ] PHP, .NET, and SQL
- [x] HTML, CSS, and DOM
- [ ] Python, Perl, and C++
- [ ] Java, ASP, and C#

**Q6. What is this code an example of?**

```
<x/>
```

- [ ] null element

- [x] self-closing tag
- [ ] improperly named element
- [ ] incorrect XML syntax

**Q7. Which XHTML syntax rule does NOT apply to XML?**

- [ ] XHTML attribute values must be quoted
- [x] XHTML tags and attributes must be in lowercase
- [ ] XHTML elements must be properly nested within each other.
- [ ] XHTML tags must have an equivalent closing tag.

**Explanation**: XML Attributes values must be quoted. Element names are case-sensitive (and CamelCase is actually one of the naming styles).

**Q8. Which Ajax method is used to exchange data with a server, using a modern browser?**

- [ ] request-XML
- [x] XMLHttpRequest
- [ ] ActiveXObject
- [ ] responseXML

**Q9. A markup language is a _ -readable language that _ text so that the computer can _ that text.**

- [ ] processor; complies; process
- [ ] system; stores; retrieve
- [ ] non; processes; format
- [x] human; annotates; manipulate

**Q10. What is this code an example of?**

```
<x a="x" a="y"></x>
```

- [x] improperly named element
- [ ] self-closing tag
- [ ] null element
- [ ] incorrect XML syntax

**Q11. XML provides a framework for specifying markup languages, while HTML is a predefined markup language. What is applicable to XML and not HTML?**

- [x] It is mandatory to use closing tags with XML
- [ ] It is important for an XML document to be well formed
- [ ] XML elements start with an opening tag in angle brackets, such as `<p>`
- [ ] XML syntax uses tags, elements, and attributes

**Q12. What is the last step in extending XHTML modules?**

- [ ] The last step is to complete the extension of XHTML compound documents and make sure the documents adhere to the defined namespaces.
- [x] The last step is to create the DTD for the XHTML extension, which references both the XHTML modules and the new modules.
- [ ] The last step is to run the XHTML extension through the XSLT processor, which will properly format it.
- [ ] The last step is to verify that the XHTML is well formed and valid, and compatible with most browsers.

**Q13. In an XML DTD ATTLIST declaration, which default value is used to indicate that the attribute does not have to be included?**

- [x] `#DEFAULT`
- [ ] `#OPTIONAL`
- [ ] `#IMPLIED`
- [ ] `#FIXED`

**Q14. How does the XML DOM present an XML document?**

- [ ] as a set of objects
- [x] as a tree structure
- [ ] as an array of nodes
- [ ] as a dynamic program

**Q15. You are working with an XML document that uses an XML schema. How do you specify that an element can appear multiple times inside its parent element?**

- [ ] Set the maxOccurs attribute to a large number, such as 1.000

- [ ] Set the maxOccurs attribute to 0
- [ ] Set the maxOccurs attribute to undefined.
- [x] Set the maxOccurs attribute to unbounded.

**Q16. The `<xsl:with-param>` element defines the value of a parameter to be passed into a template. It can be used within which elements?**

- [x] `<xsl:apply-templates>` and `<xsl:call-template>`
- [ ] `<xsl:param>` and `<xsl:processing-instruction>`
- [ ] `<xsl:template>` and `<xsl:transform>`
- [ ] `<xsl:include>` and `<xsl:variable>`

**Q17. You are checking someone else's XML document for errors. You notice that the prolog does not have a closing tag. What do you do?**

- [ ] Remove the prolog to make sure that the XML document will be properly processed across all platforms.
- [x] Leave it alone, because the prolog does not require a closing tag.
- [ ] Move the prolog to an external file so that the XML document only has elements with closing tags.
- [ ] Add a closing tag, as al XML elements must have a closing tag.

**Q18. Which statement is not true about XML?**

- [ ] XML is flexible and customizable.
- [ ] XML can be used to store data.
- [ ] XML is independent of Operating System.
- [x] XML is a replacement for HTML.

**Q19. In an XML DTD ATTLIST declaration, which tokenized attribute type is used to specify multiple ID values?**

- [ ] ENTITIES
- [x] IDREFS
- [ ] IDS
- [ ] IDSETS

**Q20. You want to convert a large XML file into CSV format. You did not create the XML file, so you are not familiar with all of the syntax. What will help you get the best insight into the file contents?**

- [ ] XSLT
- [ ] DOM
- [ ] AJAX
- [x] XSD

This question is about understanding the XML afile contents. XSD is the correct one here - that's the schema document, which describes the XML.

**Q21. In an XML DTD, attributes are declared with an ATTLIST declaration. You need to validate the color attribute for element `<car>` against a fixed list of values. Which is the correct declaration?**

- [ ] `<!ATTLIST car color (red|white|blue|black) black>`
- [ ] `<!ATTLIST car color (red|white|blue|black) #REQUIRED>`
- [x] `<!ATTLIST car color (red|white|blue|black) #FIXED>`
- [ ] `<!ATTLIST car color (red|white|blue|black)>`

**Q22. The main ways to control the display of XML documents are with Cascading Style Sheets (CSS) and Extensible Styles Language (XSL). What is an advantage of CSS over XSL?**

- [ ] CSS is a complete programming language with more powerful syntax.
- [ ] With CSS, the same element can be processed multiple times.
- [ ] CSS allows you to reformat data into completely new structures.
- [x] CSS is easier to learn, use, and maintain.

**Q23. Which type of DTD declaration is this code an example of? `<!DOCTYPE abc SYSTEM "file/file.dtd">`**

- [ ] Linked
- [ ] Internal
- [x] External
- [ ] Structured

**Q24. The purpose of an XML schema is to define the building blocks of an XML document. Which option best describes the building blocks of an XML document?**

- [ ] Header files, function declarations, global variables with their data types, and system library folder location.
- [ ] Namespace declaration, processor type, markup references, and encoding specification.
- [x] The document's elements and attributes, their data types and default values, and the number and order of child elements.
- [ ] XML entity definitions, XSLT and cascading style sheets, DOM specification, and CDATA assignments.

reference link:

**Q25. An XHTML document type definition (DTD) describes the allowed syntax and grammar of XHTML markup. Which is not one of the formal DTDs used in XHTML 1.0?**

- [ ] Frameset
- [ ] Transitional
- [x] Basic
- [ ] Strict

**Explanation**: XHTML - Doctypes

**Q26. You are working with the following XML code snippet. You have this line in your XSLT code `xsl:value-of-select="//car/make"/>`. What does it display?**

```
<cars>
    <car>
        <make>Cadillac
            <model>Escalade</model>
            <price year="2007">$20,000</price>
        </make>
    </car>
</cars>
```

- [ ] Cadillac
- [ ] Cadillac Escalade
- [ ] Cadillac Escalade 20000
- [x] Cadillac Escalade $20,000

**Q27. You need to display the list of cars in the code snippet below in a column format, with a counter column for each row. Which XPath function do you use for the counter?**

```
<cars>
```

```
    <car><make>Cadillac</make> <model>Escalade</model> <year>2007</year></car>
    <car><make>Ford</make> <model>Mustang</model> <year>1968</year></car>
    <car><make>Mercedes</make> <model>C-Class</model> <year>1999</year></car>
</cars>
```

- [ ] format-number()
- [ ] id()
- [ ] count()
- [x] position()

**Explanation**: count() returns the total the number of nodes (3), while position() returns the 0-based index of each node.

**Q28. You are working with this XML code snippet from the XML document cars.xml. You need to return the information about the cars built after the year 2000, as an ordered list, starting with the most recent. What does your XQuery look like?**

```
<cars>
    <car><make>Cadillac</make> <model>Escalade</model ><year>2007</year></car>
    <car><make>Cadillac</make> <model>Escalade</model> <year>2011</year></car>
    <car><make>Ford</make> <model>Mustang</model> <year>1968</year></car>
    <car><make>Ford</make> <model>Mustang</model> <year>1998</year></car>
    <car><make>Mercedes</make> <model>C-Class</model> <year>1999</year></car>
    <car><make>Mercedes</make> <model>C-Class</model> <year>2009</year></car>
</cars>
```

- [ ]

```
<ul>
{
    for $x in doc("cars.xml")/cars/car
    where $x/year>2000
    order by $x/year descending
    return <li>{$x}</li>
}
</ul>
```

- [ ]

```
<ol>
{
    for $x in doc("cars.xml")/cars/car
    where $x/year>2000
    order by $x/year desc
    return <li>{data($x)}</li>
}
</ol>
```

- [ ]

```
<ul>
{
    for $x in doc("cars.xml")/cars/car
    where $x/year>2000
    order by $x/year
    return <li>{$x}</li>
}
</ul>
```

- [x]

```
<ol>
{
    for $x in doc("cars.xml")/cars/car
    where $x/year>2000
    order by $x/year descending
    return <li>{data($x)}</li>
}
</ol>
```

reference link:

**Q29. The `readyState` property holds the status of the XMLHttpRequest. Which is NOT a valid status?**

- [ ] `4 (DONE)`
- [ ] `3 (LOADING)`
- [x] `1 (PROCESSING)`
- [ ] `0 (UNSENT)`

reference link:

**Q30. You are working with an XML document that uses an XML schema. How can you extend the document with elements NOT specified by the schema?**

- [x] Use the `<any>` element.
- [ ] Use the `<redefine>` element.
- [ ] Use `<xs:extension>`.
- [ ] Specify the new elements in the schema.

reference link:

**Q31. You are working with the following XML code snippet. Which XPath expression produces C-Class?**

```
<cars>
    <car><make>Cadillac</make><model>Escalade</model>
        <price year="2007">20000</price></car>
    <car><make>Ford</make><model>Mustang</model>
```

```
        <price year="2008">17000</price></car>
    <car><make>Mercedes</make><model>C-Class</model>
        <price year="2009">24000</price></car>
</cars>
```

- [ ] `/car[price>20000]/make/model`
- [ ] `/car[price>=20000 and @year>=2009]/make/model`
- [ ] `//car[price>=20000 and @year>2008]/model`
- [ ] `/cars/car[price>=20000 and year>2008]/model`

NOTE: XPather shows that all answers are incorrect. Report the question.

**Q32. You are working with an XML document that uses an XML schema. How do you ensure that an attribute must be specified for its corresponding element?**

- [ ] Set the type attribute to `xs:required`.
- [x] Set the `use` attribute to `required`.
- [ ] Set the minLength attribute to 1.
- [ ] Set the minOccurs attribute to 1.

**Reference**: XSD Attributres

**Q33. You are working with the following XML code snippet. What do you need to include in your XSLT code to display `Mercedes, Cadillac, Ford`?**

```
<cars>
    <car><make>Cadillac</make><model>Escalade</model>
        <price year="2007">20000</price></car>
    <car><make>Ford</make><model>Mustang</model>
        <price year="2008">17000</price></car>
    <car><make>Mercedes</make><model>C-Class</model>
        <price year="2009">24000</price></car>
</cars>
```

- [ ] `<xsl:sort select="make" />`
- [x] `<xsl:sort select="model" />`
- [ ] `<xsl:sort select="car" />`
- [ ] `<xsl:sort select="price" />`

**Explanation**: A trick question. The `<xsl:sort>` will sort the output in ascending (alphabetical for strings) order by default. The `select` tells which tag to use for sorting.

- If we use `select="make"` or `select="year"` we get the order `Cadillac, Ford, Mercedes`

- If we use `select="price"` we get `Ford, Cadillac, Mercedes`
- And finally with `select="model"` we get `Mercedes, Cadillac, Ford`

**Q34. What is the correct syntax for comments in XQuery?**

- [ ] `/* */`
- [ ] `<!-- -->`
- [ ] `//`
- [x] `(: :)`

**Q35. Which DOM node type may NOT have the EntityReference node type as one of its child nodes?**

- [ ] Element
- [x] Document
- [ ] EntityReference
- [ ] DocumentFragment

reference link:

**Q36. XHTML modules can be extended by adding elements, attributes, modifying content models, or some combination of these. What does a proper implementation of an XHTML module require?**

- [ ] The implementation of an XHTML module requires an extension module and a validation module that ensures that the XHTML is well formed and valid; otherwise the extended instances aren't formally XHTML.
- [ ] The implementation of an XHTML module requires a definitions module and a constraint module that specifies syntax rules and uses the parameter entities declared in the definitions module.
- [ ] The implementation of an XHTML module requires a qualified name module and a declaration module that holds the element, element attribute, and content model declarations.
- [x] The implementation of an XHTML module requires a namespace module that holds the element, element attribute, and content model declarations, and a parameter module that uses the entities declared in the namespace module.

**Q37. The `<xsl:namespace-alias>` element is used to replace a namespace in the style sheet with a different namespace in the output. Which XSLT element needs to be its parent node?**

- [ ] `<xsl:namespace>`
- [ ] any valid element
- [x] root element
- [ ] top-level element in the corresponding namespace

reference link:

**Q38. XML is a markup language, not a programming language. What makes XML not qualify to be a programming language?**

- [ ] XML is too flexible and does not have enough reserved keywords.
- [x] XML contains only data and not any processing instructions.
- [ ] XML does not perform any computation or algorithms.
- [ ] XML does not have specialized syntax rules.

**Q39. What is true about these elements in XQuery?**

```
<cars>
    <car><make>Cadillac</make><model>Escalade</model><year>2007</year></car>
    <car><make>Cadillac</make><model>Escalade</model><year>2011</year></car>
    <car><make>Ford</make><model>Mustang</model><year>1968</year></car>
    <car><make>Ford</make><model>Mustang</model><year>1998</year></car>
    <car><make>Mercedes</make><model>C-Class</model><year>1999</year></car>
    <car><make>Mercedes</make><model>C-Class</model><year>2009</year></car>
</cars>
```

- [ ] Elements `<make>` and `<model>` are ancestors of `<year>`.
- [ ] Elements `<make>` and `<model>` are children of `<cars>`.
- [x] Elements `<make>` and `<model>` are siblings.
- [ ] Elements `<car>` and `<cars>` are parents of `<make>` and `<model>`.

**Q40. Which is a valid CSS section for this XML code snippet?**

```
<cars>
    <car><make>Cadillac</make><model>Escalade</model><year>2007</year></car>
    <car><make>Ford</make><model>Mustang</model><year>1968</year></car>
    <car><make>Mercedes</make><model>C-Class</model><year>1999</year></car>
</cars>
```

- [ ]

```
cars {
  display: block;
}
car(make),
car(model),
car(year) {
  display: inline;
  padding-top: 0.5em;
}
```

- [x]

```
car, cars { display: block; }
make, model, year {
  display: inline;
  padding-top: 0.5em;
}
```

- [ ]

```
cars {
  display: block;
}
car.make,
car.model,
car.year {
  display: inline;
  padding-top: 0.5em;
}
```

- [ ]

```
cars {
  display: block;
}
car#make,
car#model,
car#year {
  display: inline;
  padding-top: 0.5em;
}
```

**Q41. An XML document contains this code as part of the DTD: . What are the rules that need to be followed for each of the elements?**

- [ ] is required, is optional, is optional, and is optional.
- [ ] is required, is required, is optional, and is optional.
- [ ] is required, is required, is required, and is optional.
- [x] is required, is optional, is required, and is optional.

**Q42. Which element in this XML code is not a good candidate for conversion into an attribute?**

```
1 <superheroes>
2 <name>Superman</name>
3 <alias>Clark Kent</alias>
4 <birthplace>Krypton</birthplace>
5 <power>Flight</power>
6 <power>X-Ray Vision</power>
7 <power>Super Strength</power>
8 </superheroes>
```

- [ ] `<birthplace>`
- [ ] `<alias>`
- [ ] `<name>`
- [x] `<power>`

Reference best practices for xml attributes

**Q43. What does the Document Type Definition (DTD) define?**

- [ ] structure
- [ ] entities
- [x] elements
- [ ] attributes

**Q44. In the XML DOM,what is the `setAttribute()` an example of?**

- [ ] node
- [ ] function
- [x] method
- [ ] property

**Q45. What is not one of the advantages of the XML DOM?**

- [ ] The XML DOM is language and platform independent.
- [ ] The XML DOM is modifiable and dynamic.
- [ ] The XML DOM is easy to navigate around to find specific information.
- [x] The XML DOM is efficient with memory and operation speed.

**Q46. In the XML DOM, which property is best to use to loop through each of the nodes i nthe code snippet below?**

```
<cars>
    <car><make>Cadillac</make><model>Escalade</model><year>2007</year></car>
    <car><make>Ford</make><model>Mustang</model><year>1968</year></car>
    <car><make>Mercedes</make><model>C-Class</model><year>2006</year></car>
</cars>
```

- [ ] nextChild
- [ ] nextSibling
- [x] nodeValue
- [ ] nodename