



< Previous



Next >

Notifications



XML Quiz

Bookmark this page

Pursue a verified certificate

- Earn a **verified certificate** of completion to showcase on your resumé
- Support our **mission** at edX

Upgrade for \$50



Hide Notes

Quiz due Apr 29, 2022 08:53 IST

Each multiple-choice quiz problem is based on a "root question," from which the system generates different correct and incorrect choices each time you take the quiz. Thus, you can test yourself on the same material multiple times. We strongly urge you to continue testing on each topic until you complete the quiz with a perfect score at least once. Simply click the "Reset" button at the bottom of the page for a new variant of the quiz.

After submitting your selections, the system will score your quiz, and for incorrect answers will provide an "explanation" (sometimes for correct ones too). These explanations should help you get the right answer the next time around. To prevent rapid-fire guessing, the system enforces a minimum of 10 minutes between each submission of solutions.

Q1

1.0/1.0 point (graded)

[Q1] We're interested in well-formed XML that satisfies the following conditions:

- It has a root element "tasklist"
- The root element has 3 "task" subelements
- Each of the "task" subelements has an attribute named "name"
- The values of the "name" attributes for the 3 tasks are "eat", "drink", and "play"

Select, from the choices below, the well-formed XML that meets the above requirements.

☐

```
<tasklist>
  <task name=eat/>
  <task name=drink/>
  <task name=play/>
</tasklist>
```

☐

```
<tasklist>
  <task name="eat">
  <task name="drink">
  <task name="play">
</tasklist>
```

☐

```
<tasklist>
  <task name="eat"/>
</tasklist>
<tasklist>
  <task name="drink"/>
</tasklist>
```

```
<tasklist>
  <task name="play"/>
</tasklist>
```



```
<tasklist>
  <task name="eat"></task>
  <task name="drink"></task>
  <task name="play"></task>
</tasklist>
```



Problem Explanation

Well-formed XML must follow these rules (along with others):

- There must be exactly one top level element.
- All opening tags must be closed.
- All elements are properly nested i.e., there are no interleaved elements.
- Attribute values must be enclosed in single or double quotes.

Submit

i Answers are displayed within the problem

Q2

1.0/1.0 point (graded)

[Q2] An XML document contains the following portion:

```
<INFO>
  <ADDR>101 Maple St.</ADDR>
  <PHONE>555-1212</PHONE>
  <PHONE>555-4567</PHONE>
</INFO>
```

Which of the following could be the INFO element specification in a DTD that the document matches?



<!ELEMENT INFO (ADDR?,PHONE+)>



<!ELEMENT INFO (ADDR*,PHONE+,MANAGER)>



<!ELEMENT INFO (#PCDATA)>



<!ELEMENT INFO (ADDR,PHONE)>



Problem Explanation

In the XML snippet, the info element has one address subelement and two phone subelements, in that order. Thus, in the DTD the list of components for INFO must include ADDR, ADDR*, ADDR+, or ADDR? followed by PHONE*, PHONE+, or PHONE? in that order.

Hide Notes

ADDR? followed by PHONE* or PHONE+. Interspersed with these may be any elements that are not required to appear-- that is, any components with a ? or *. Thus, we might also have components like NAME* or MANAGER? at any point in the list.

Submit

i Answers are displayed within the problem

Q3

1.0/1.0 point (graded)

[Q3] An XML document contains the following portion:

```
<EMP name = "Kermit">
  <ADDR>123 Sesame St.</ADDR>
  <PHONE type = "cell">555-1212</PHONE>
</EMP>
```

Which of the following could NOT be part of a DTD that the document matches? Note that there can be multiple ATTLIST declarations for a single element type; do not assume the only attributes allowed for an element type are the ones shown in the answer choice.

☒ <!ATTLIST PHONE owner IDREF #REQUIRED>

☐ <!ATTLIST PHONE type IDREF #IMPLIED>

☐ <!ATTLIST EMP name ID #IMPLIED>

☐ <!ATTLIST EMP ssNo ID #IMPLIED>



Answer-Selection Feedback

This is a correct choice, because a #REQUIRED attribute must appear with every element. In the example fragment of a document, we do not see the owner attribute in the PHONE element.

Problem Explanation

The correct choices (i.e., the erroneous DTD snippets) are based on two rules:

1. A #REQUIRED attribute must appear in every element.
2. An attribute can have types CDATA, ID, or IDREF(S), but not #PCDATA.

The incorrect choices (i.e., the snippets that *could* appear in a DTD), are either optional attributes (#IMPLIED) or are required attributes of a proper type.

Submit

i Answers are displayed within the problem

Hide Notes

Q4

1/1 point (graded)

[Q4] Here is a DTD:

```
<!DOCTYPE A [  
  <!ELEMENT A (B+, C)>  
  <!ELEMENT B (#PCDATA)>  
  <!ELEMENT C (B?, D)>  
  <!ELEMENT D (#PCDATA)>  
>
```

Which of the following sequences of opening and closing tags matches this DTD? Note: In actual XML, opening and closing tags would be enclosed in angle brackets, and some elements might have text subelements. This quiz focuses on the element sequencing and interleaving specified by the DTD.

☐ A B /B B /B C B /B D /D D /D /C /A

☐ A B /B B /B C B /B /C /A

☒ A B /B B /B C D /D /C /A

☐ A B /B B /B C /C /A



Problem Explanation

According to the DTD, an A element has within it one or more B subelements, and then a C element. Within the C element is zero or one B elements followed by exactly one D element. In terms of regular expressions, the tag sequences we can see are:

A (B /B)(B /B)* C (D /D | B /B D /D) /C /A.

Some text may appear between each B-/B pair and each D-/D pair, but text may not appear elsewhere.

Submit

i Answers are displayed within the problem

Q5

1.0/1.0 point (graded)

[Q5] Here is an XML DTD:

```
<!DOCTYPE meal [  
  <!ELEMENT meal (person*, food*, eats*)>  
  <!ELEMENT person EMPTY>  
  <!ELEMENT food EMPTY>  
  <!ELEMENT eats EMPTY>  
  <!ATTLIST person name ID #REQUIRED>  
  <!ATTLIST food name ID #REQUIRED>  
  <!ATTLIST eats diner IDREF #REQUIRED dish IDREF  
#REQUIRED>  
>
```

Hide Notes

Which of the following documents match the DTD?

1.

```
<meal>
  <person name="Alice"/>
  <food name="salad"/>
  <eats diner="Alice" dish="salad"/>
  <person name="Bob"/>
  <food name="salad"/>
  <eats diner="Bob" dish="salad"/>
  <person name="Carol"/>
  <food name="sandwich"/>
  <eats diner="Carol" dish="sandwich"/>
</meal>
```

2.

```
<meal>
  <person name="Alice"/>
  <person name="Bob"/>
  <person name="Carol"/>
  <person name="Dave"/>
  <food name="salad"/>
  <food name="turkey"/>
  <food name="sandwich"/>
  <eats diner="Alice" dish="turkey"/>
  <eats diner="Bob" dish="salad"/>
  <eats diner="turkey" dish="Dave"/>
</meal>
```

3.

```
<meal>
  <person name="Alice"/>
  <person name="Bob"/>
  <food name="salad"/>
  <eats diner="Alice" dish="food"/>
  <eats diner="Bob" dish="food"/>
</meal>
```

☐ only the third

☐ only the first

☐ none of the three

☒ only the second



Problem Explanation

Focus on the ID and IDREF attributes: A valid document needs to have unique values across ID attributes. An IDREF attribute can refer to any existing ID attribute value.

Submit

i Answers are displayed within the problem

Q6

1.0/1.0 point (graded)

[Q6] Study the following XML Schema specification:

Hide Notes

[Q0] Study the following XML Schema specification.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="person">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="fname" type="xs:string"/>
        <xs:element name="initial" type="xs:string"
          minOccurs="0"/>
        <xs:element name="lname" type="xs:string"/>
        <xs:element name="address" type="xs:string"
          maxOccurs="2"/>
        <xs:choice>
          <xs:element name="major" type="xs:string"/>
          <xs:element name="minor" type="xs:string"
            minOccurs="2" maxOccurs="2"/>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Select, from the choices below, the XML that is valid according to the XML Schema specification above.



```
<person>
  <fname>John</fname>
  <initial>Q.</initial>
  <lname>Public</lname>
  <major>History</major>
  <major>Psychology</major>
</person>
```



```
<person>
  <fname>John</fname>
  <initial>Q</initial>
  <lname>Public</lname>
  <address>123 Public Avenue, Seattle, WA
98001</address>
  <minor>Psychology</minor>
  <minor>History</minor>
</person>
```



```
<person>
  <fname>John</fname>
  <initial>Q.</initial>
  <lname>Public</lname>
  <address>123 Public Avenue</address>
  <address>Seattle</address>
  <address>WA 98001</address>
</person>
```



```
<person>
  <fname>J.</fname>
  <lname>Public</lname>
  <title>Sr.</title>
  <major>Computer Science</major>
```

```
<address>123 Public Avenue</address>
<address>Seattle</address>
<address>WA 98001</address>
</person>
```



Problem Explanation

This question deals with the `xs:element`, `xs:sequence`, and `xs:choice` elements in XML Schema. In order for XML to be valid according to the specified schema:

- The elements contained in a sequence must appear in exactly the same order as specified in the `xs:sequence`.
- Exactly one of the elements contained in an `xs:choice` must appear.
- If an element specifies a `minOccurs` attribute, the XML must contain at least that many instances of the element.
- If an element specifies a `maxOccurs` attribute, the XML must not contain more than that many instances of the element.
- If `minOccurs` and `maxOccurs` are not specified, their default value is 1.
- Elements not defined as a part of a sequence or choice cannot occur inside the corresponding `xs:sequence` and `xs:choice`.

The given schema specifies the following constraints:

- The "fname", "initial", "lname", and "address" elements must

[< Previous](#)[Next >](#)

edX

[About](#)[Affiliates](#)[edX for Business](#)[Open edX](#)[Careers](#)[News](#)

Legal

[Terms of Service & Honor Code](#)[Privacy Policy](#)[Accessibility Policy](#)[Hide Notes](#)

Connect

- [Blog](#)
[Contact Us](#)
[Help Center](#)
[Media Kit](#)
-



© 2022 edX LLC. All rights reserved.
深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)