1. If you were to assign an element a unique value through an attribute, what data type would you assign to that attribute in the DTD? ________________  

2. Will the XML document in listing 1.1 load without errors in IE6? [y/n] ______  

3. Will the XML document in listing 1.2 load without errors in IE6? [y/n] ______  

4. Which of these DTD fragments is valid? ______  
   a. <!ELEMENT paragraph (emphasis | crossref | #PCDATA)+>  
   b. <!ELEMENT paragraph (crossref | #PCDATA | emphasis)+>  
   c. <!ELEMENT paragraph (#PCDATA | emphasis | crossref)+>  
   d. none of the above  

5. Which of the following is not possible using CSS with an XML document? ______  
   a. Formatting the contents of an element in bold.  
   b. Creating a table with headers not present in the XML document  
   c. Creating a table with data in XML document  
   d. All of the above  

6. For the XML document in listing 2, write an XSLT template that prints in bold every second LINE in a STANZA that contains the word “human”. (hint: use and in the match predicate)
7. Consider the XSLT document in listing 3. Which of the templates can be deleted from the document without affecting the formatted output? (you may encircle the appropriate templates on the listing with your pencil.)

8. The result elements of an XSLT transform must always belong to XHTML (XML compliant HTML). [y|n] ____________

9. Consider the following XML Schema fragment.

```xml
<xsd:element name="checkbook">
  <xsd:complexType>
    <xsd:all>
      <xsd:element ref="transactions">
        <xsd:element ref="balance" minOccurs="0" maxOccurs="1">
          <xsd:element ref="accountholdername">
        </xsd:element>
      </xsd:element>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Which of the following DTD fragments is this equivalent to? ____________

- a. `<!ELEMENT checkbook (transactions*, balance, accountholdername)>
- b. `<!ELEMENT checkbook (transactions, balance?, accountholdername)>
- c. `<!ELEMENT checkbook (transactions | balance | accountholdername)+>
- d. none of the above

10. Name the mechanism that allows an XML schema document to contain more than one element with the same name. ________________

11. For the XSD document in listing 4, what are the URIs for the:

   a. Default Namespace: ________________
   b. Target Namespace: ________________
<?xml version="1.0"?>
<pizza>
  <price> $13.00 </price>
  <topping> onion </topping>
  <size> medium </size>
</pizza>

Listing 1.1

<?xml version="1.0"?>
<!ELEMENT pizza (price, topping, size)>

Listing 1.2

<POEM>
  <TITLE>Lines Written in Early Spring</TITLE>
  <AUTHOR>
    <FIRSTNAME>William</FIRSTNAME>
    <LASTNAME>Wordsworth</LASTNAME>
  </AUTHOR>
  <STANZA>
    <LINE N="1">I heard a thousand blended notes,</LINE>
    <LINE N="2">While in grove I sate reclined,</LINE>
    <LINE N="3">In that sweet mood when pleasant thoughts</LINE>
    <LINE N="4">Bring sad thoughts to the mind.</LINE>
  </STANZA>

  <STANZA>
    <LINE N="5">To her fair works did nature link</LINE>
    <LINE N="6">The human soul that through me ran;</LINE>
    <LINE N="7">And much it griev'd me my heart to think</LINE>
    <LINE N="8">What man has made of man.</LINE>
  </STANZA>
</POEM>

Listing 2
Listing 3
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.cdcatalog.org"
    xmlns:cd="http://www.cdcatalog.org"
    elementFormDefault="qualified">
  <element name="CD">
    <complexType>
      <sequence>
        <element ref="cd:TITLE"/>
        <element ref="cd:COUNTRY"/>
        <element ref="cd:COMPANY"/>
        <element ref="cd:YEAR"/>
        <element ref="cd:COVER_IMAGE"/>
      </sequence>
      <attribute ref="cd:artist" use="required"/>
      <attribute ref="cd:price" use="required"/>
    </complexType>
  </element>
  <attribute name="artist" type="string"/>
  <attribute name="price" type="money"/>
  <simpleType name="money">
    <restriction base="string">
      <pattern value="(\$)?\d{1,2}\.(\d{1,2})"/>
    </restriction>
  </simpleType>
  <element name="TITLE" type="string"/>
  <element name="COUNTRY" type="string"/>
  <element name="COMPANY" type="string"/>
  <element name="YEAR" type="gYear"/>
  <element name="COVER_IMAGE">
    <complexType>
      <attribute ref="cd:source" use="required"/>
    </complexType>
  </element>
  <attribute name="source" type="anyURI"/>
</schema>
Listing 4