

Problem	1	2	3	Total
Points:	40	10	30	80
Score:				

This homework assignment has 3 problems, for a total of 80 points.

1. This problem deals with architectures.
 - (a) (8 points) Identify which of the following statements about three-tier architecture are true
 - A. The main payoff of three-tier architectures is in improving system development and management
 - B. Three-tier architectures separate presentation, business logic, and data access from each other
 - C. Three-tier architectures make no sense unless you have installed DBMSs from at least two vendors
 - D. When mobile computing becomes more prevalent in enterprises, three-tier architectures will need to be replaced by four-tier architectures
 - (b) (8 points) Message-oriented middleware (mark all that are true)
 - A. Guarantees reliable delivery of messages
 - B. Can only be implemented via a minimum of three routers to route messages from a sender to a receiver
 - C. Guarantees reliable delivery of messages or a failure notification to the sender
 - D. Enables one receiver to subscribe to more than one topic
 - (c) (8 points) Of the following statements, identify all that are true about open or closed environments:
 - A. When implementing a large system, it is often advisable to treat a open environment as closed
 - B. When implementing a large system, it is often advisable to treat a closed environment as open
 - C. The set of components in an open environment can change (almost) arbitrarily
 - D. Open environments presuppose the use of message-oriented middleware over which XML documents are exchanged
 - (d) (4 points) The main ingredients of an architecture are
 - A. Components and environments
 - B. Components, organizations, and environments
 - C. Interconnections, separations, and environments
 - D. Components and interconnections
 - (e) (12 points) Give an example for each of the following (enterprise) architecture modules.
 - i. Example applications module:
 - ii. Example systems module:
 - iii. Example infrastructure module:

2. (10 points) List any three of the main uses of XML (in about 15 words total).

Listing 1: Unique songs nested in unique singers

3.

```
<Songs>
  <Sgr name="Eagles" genre="rock">
    <Song lg="en">Hotel California</Song>
    <Song lg="en">Seven Bridges Road</Song>
  </Sgr>
  <Sgr name="H_Belafonte" genre="reggae">
    <Song lg="cpe">Day O</Song>
    <Song lg="en">Jamaica Farewell</Song>
  </Sgr>
</Songs>
```

Mark the appropriate choices to complete the following XML Schema snippets for Listing 1. Ignore the missing components and ignore namespaces.

```
<xsd:element name="Songs" type="SongsT"/>

<xsd:complexType name="SongsT">
  <xsd:sequence>
    <xsd:element name="Sgr" type="SgrT" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<element name="Sgr" type="SgrT"/>

<complexType name="SgrT"> <!-- PART (a) -->
</complexType>

<complexType name="SongT"> <!-- PART (b) -->
</complexType>
```

(a) (15 points) The definition of SgrT should be

- A.
- ```
<sequence>
 <element name="Song" type="SongT"/>
</sequence>
<attribute name="name" type="string"/>
```
- B.
- ```
<sequence>
```

```
<element name="Song" type="SongT" maxOccurs="unbounded"/>
</sequence>
<attribute name="name" type="string"/>
```

C.

```
<sequence>
  <element ref="Song" type="SongT" maxOccurs="unbounded"/>
  <attribute name="name" type="string"/>
</sequence>
```

D.

```
<sequence>
  <element ref="Song" type="SongT" maxOccurs="unbounded"/>
</sequence>
<attribute name="name" type="string"/>
```

(b) (15 points) The definition of SongT should be

E.

```
<sequence>
  <attribute name="lg" type="lgT"/>
</sequence>
```

F.

```
<sequence>
</sequence>
<attribute name="lg" type="lgT"/>
```

G.

```
<simpleContent>
  <attribute name="lg" type="lgT"/>
</simpleContent>
```

H.

```
<simpleContent>
  <extension base="string">
    <attribute name="lg" type="lgT"/>
  </extension>
</simpleContent>
```