

Exercise 1 Suggest a DTD for the XML file in Figure A

FIGURE A:

```
<?xml version="1.0" encoding="UTF-8" ?>

<volunteerDatabase>
  <person age="31" ssn="046187254">
    <name>
      <firstname>Ross</firstname>
      <lastname>Geller</lastname>
    </name>
    <telephone type="landline">
      <number>5534567</number>
    </telephone>
    <telephone type="mobile">
      <number>0851234567</number>
    </telephone>
  </person>

  <person age="29" ssn="355817204">
    <name>
      <firstname>Chandler</firstname>
      <firstname>Muriel</firstname>
      <lastname>Bing</lastname>
    </name>
    <telephone type="mobile">
      <number>0869932617</number>
    </telephone>
  </person>

  <person ssn="778123666">
    <name>
      <firstname>Joseph</firstname>
      <firstname>Francis</firstname>
      <lastname>Tribbiani</lastname>
    </name>
    <telephone type="landline">
      <number>01628777</number>
    </telephone>
  </person>
</volunteerDatabase>
```

EXAMPLE DTD to show SYNTAX

```
<!DOCTYPE NEWSPAPER [

  <!ELEMENT NEWSPAPER (ARTICLE+)>
  <!ELEMENT ARTICLE
    (HEADLINE,BYLINE+,LEAD?,BODY,NOTES*)>
  <!ELEMENT HEADLINE (#PCDATA)>
  <!ELEMENT BYLINE (#PCDATA)>
  <!ELEMENT LEAD (#PCDATA)>
  <!ELEMENT BODY (#PCDATA)>
  <!ELEMENT NOTES (#PCDATA)>

  <![ATTLIST ARTICLE AUTHOR CDATA #REQUIRED]
  <![ATTLIST ARTICLE EDITOR CDATA #IMPLIED]
  <![ATTLIST ARTICLE DATE CDATA #IMPLIED]
  <![ATTLIST ARTICLE EDITION CDATA #IMPLIED]

  <![ENTITY NEWSPAPER "Trinity Times"]
  <![ENTITY PUBLISHER "Trinity Press"]
  <![ENTITY COPYRIGHT "Copyright 1998 TCD Press"]

]>
```

- Syntax for occurrences of elements in DTDs
 - ? : zero-or-one
 - + : one-or-more
 - * : zero-or-more

Exercise 2: Design XPath queries for

1. Get all the firstnames of persons in the file (without using //)
2. Get just the text from the firstname elements of name
3. Return only the person elements that has an age attribute
4. Return the entire person element whose firstname is "Chandler"
5. Return all the values of ssn attributes
6. Return the number of each person who has an age attribute

FIGURE A:

```
<?xml version="1.0" encoding="UTF-8" ?>

<volunteerDatabase>
  <person age="31" ssn="046187254">
    <name>
      <firstname>Ross</firstname>
      <lastname>Geller</lastname>
    </name>
    <telephone type="landline">
      <number>5534567</number>
    </telephone>
    <telephone type="mobile">
      <number>0851234567</number>
    </telephone>
  </person>

  <person age="29" ssn="355817204">
    <name>
      <firstname>Chandler</firstname>
      <firstname>Muriel</firstname>
      <lastname>Bing</lastname>
    </name>
    <telephone type="mobile">
      <number>0869932617</number>
    </telephone>
  </person>

  <person ssn="778123666">
    <name>
      <firstname>Joseph</firstname>
      <firstname>Francis</firstname>
      <lastname>Tribbiani</lastname>
    </name>
    <telephone type="landline">
      <number>01628777</number>
    </telephone>
  </person>
</volunteerDatabase>
```

Exercise 3

Explain using examples what constitutes a well formed and valid XML document.

- **Well formed-** XML Declaration required,
- Exactly one root element,
- Empty elements are written in one of two ways: Closing tag or Special start tag, For non-empty elements, closing tags are required,
- Attribute values must always be quoted,
- Start tag must match closing tag (name & case),
- Correct nesting of elements



Exercise 4

- NOW using **bib.xml** (on next slide) List books published by Addison-Wesley after 1991, including their year and title.

You should get

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
  </book>
  <book year="1992">
    <title>Advanced Programming in
the Unix environment</title>
  </book>
</bib>
```

- Example syntax

```
let $c:=
  doc("data/tcd.xml")/assessments/
course/mark
return
  <list_of_avg_course_marks>
    {$c}
  </list_of_avg_course_marks>

for $j in
  doc("data/tcd.xml")/assessments/
course/@name
return
  <one_of_courses_is>
    {$j}
  </one_of_courses_is>
```

Bib.xml

```
<?xml version="1.0" ?>
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>

  <book year="1992">
    <title>Advanced Programming in the Unix
environment</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>

  <book year="2000">
    <title>Data on the Web</title>
    <author><last>Abiteboul</last><first>Serge</first></author>
    <author><last>Buneman</last><first>Peter</first></author>
    <author><last>Suciu</last><first>Dan</first></author>
    <publisher>Morgan Kaufmann Publishers</publisher>
    <price>39.95</price>
  </book>

  <book year="1999">
    <title>The Economics of Technology and Content for
Digital TV</title>
    <editor>
      <last>Gerbarg</last><first>Darcy</first>
      <affiliation>CITI</affiliation>
    </editor>
    <publisher>Kluwer Academic Publishers</publisher>
    <price>129.95</price>
  </book>
</bib>
```

