## Practice # 1

An XML document is parsed by Internet Explorer and the Following output is displayed

```
<?xml version="1.0" ?>
- <!-- Character entities and Special characters example  -->
<examples>
    <example>Show & Tell Consulting ®</example>
    <example>©Microsoft Corporation 2000</example>
    <example>7 is > 6</example>
    <example>This is a trainee's exam result</example>
</examples>
```

Write an XML document using notepad and make sure that the document can be parsed without error.

## Practice # 2

You have the following XML file idbbisew.xml

```
<?xml version='1.0'?>
<scholarship>
    <runby>IDB-BISEW<runby>
    <objective>To improve skills of underprivileged Muslim youth in IT field<objective>
</scholarship>
```

Now declare two parsed character entities for the text phrases“IDB-BISEW” and “To improve skills of underprivileged Muslim youth in IT field”.
Rewrite the XML document referencing the two entities in the element contents.
Use internal DTD subset. You do not need to declare elements or validate the document.

## Practice # 3

You have the following sample XML file containing a IDB-BISEW scholarship course.

```
<?xml version='1.0'?>
<course course-id="ESAD-CS">
    <name>Enterprise System Analysis and Design using Visual C Sharp</name>
    <modules>
        <module moduleid="ESADCS-01">  ,
            <name>Programming Concepts</name>
            <duration>80</duration>
        </module>
        <module moduleid="ESADCS-02">
            <name>Designing and implementing databases</name>
            <duration>60</duration>
        </module>
    </modules>
</course>
```

You are also the following constraints the XML element and attributes

- course-id attribute must appear and its value must be ESAD-CS, ESAD-VB or NT
- module-id attribute is required and must unique in the documents
- modules must contain at least one module

Write a DTD file and validate the XML document.

## Practice # 4

Consider the following facts

- A person’s name consists of first-name, middle-name, and last-name in order as written

- First-name, last-name and middle-name – each one contains text data
- Some person has no middle-name
- First-name, last-name can appear only once

Tasks:
Write DTD code fragments for the above content model and specification.
Write sample XML content
Write DTD rules as internal subset in the XML document

## Practice # 5

You have to create persons Document Type. Structure is described below

- Persons
- persons contains zero one or more person elements
- person element has an optional title attribute
- title attribute may have “Mr” or “Mrs” as the value
- person element contains name element
- name element contains text data

Tasks
Create an external DTD file
Write an example XML file with sample data and associate the external DTD reference
Validate the XML document

## Practice # 6

**You want to separate your HTML user interface from content. You stored data in XML in following structure**

```
<?xml version="1.0" encoding="utf-8" ?>
<trainee>
  <name>Md. Foysal Wahid</name>
  <email>fwrasel87@gmail.com</email>
  <course>ESAD-CS</course>
  <round>18</round>
</trainee>
```

Tasks:
Create an HTML template
Show data on the HTML template using JavaScript as middle layer application

## Practice # 7

**You have the following XML data in book.xml**

```
<?xml version="1.0"?>
<book>
      <name>SQL</name>
      <price>700.00</price>
      <author>
           <first-name>James</first-name>
           <last-name>Scott</last-name>
      </author>
</book>
```

Using Java Script show the xml data on an HTML page like below Book Information
Name : SQL
Author: James Scott
Price : 700.00

## Practice # 8

**You have the following XML data in trainees.xml**

```
<?xml version="1.0"?>
<trainees>
    <trainee course="ESAD" round="11">
        <name>Nazmul Haq</name>
        <assignedto>DIIT</assignedto>
        <contact>0110889933</contact>
    </trainee>
    <trainee course ="NT" round="11">
        <name>Anwar</name>
        <assignedto>DIIT</assignedto>
        <contact>0110887766</contact>
    </trainee>
    <trainee course ="ESAD" round="11">
        <name>Sanu</name>
        <assignedto>BITL</assignedto>
        <contact>0110786094</contact>
    </trainee>
</trainees>
```

**You have to show the information of trainees in ESAD course on an HTML file using JavaScript.**

## Practice # 9

You have an xml file like below:

```
<?xml version='1.0'>
<trainee id="107898">
    <name>Kohorshed Alam</name>
    <email>alam@idb-bisew.org</email>
    <course>ESAD-CS</course>
    <round>11</round>
</trainee>
```

Create XSL rules to display XML data in HTML as below

Trainee Information

| | |
|---|---|
| Name | Kohorshed Alam |
| ID | 107898 |
| Email | alam@idb-bisew.org |
| Course | ESAD-CS |
| Round | 11 |

## ***Practice # 10***

Write the following XML file

**File:** books.xml

```
<?xml version="1.0"?>
<books>
    <book isbn="10-90-887-909">
        <name>SQL ins and outs</name>
        <author>Clerk</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>4</edition>
```

```
        <publisher>QUE</publisher>
    </book>
    <book isbn="10-90-3467-819">
        <name>OOSAD using UML</name>
        <author>Bennet</author>
        <listprice>24.99</listprice>
        <price>17.55</price>
        <edition>3</edition>
        <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819">
        <name>XML</name>
         <author>Bennet</author>
         <listprice>24.99</listprice>
         <price>17.55</price>
         <edition>2</edition>
         <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819">
         <name>HTML</name>
         <author>Bennet</author>
         <listprice>24.99</listprice>
         <price>17.55</price>
         <edition>4</edition>
         <publisher>Prentice Hall</publisher>
    </book>
</books>
```

Now create a XSL file and write appropriate XSLT rules so that a XSLT processor generate following HTML output

**Available books in store:**

| Name | Author | Price | Edition | Publisher |
|---|---|---|---|---|
| SQL ins and outs | Clerk | $17.55 | 4 | QUE |
| OOSAD using UML | Bennet | $17.55 | 3 | Prentice Hall |
| XML | Bennet | $17.55 | 2 | Prentice Hall |

## ***Practice # 11***

Write the following XML file below

**File:** books.xml

```
<?xml version="1.0"?>
<books>
    <book isbn="10-90-887-909" available="yes">
        <name>SQL ins and outs</name>
        <author>Clerk</author>
        <listprice>24.99</listprice>
        <price>10.55</price>
        <edition>4</edition>
        <publisher>QUE</publisher>
    </book>
    <book isbn="10-90-3467-819" available="no">
```

```
        <name>Programming practice</name>
        <author>Dave Hart</author>
        <listprice>20.99</listprice>
        <price>8.55</price>
        <edition>4</edition>
        <publisher>Prentice Hall</publisher>
    </book>
  <book isbn="10-90-3467-819" available="yes">
        <name>OOSAD using UML</name>
        <author>Bennet</author>
        <listprice>30.99</listprice>
        <price>20.55</price>
        <edition>3</edition>
        <publisher>Prentice Hall</publisher>
    </book>
    <book isbn="10-90-3467-819" available="no">
        <name>HTML</name>
        <author>Bennet</author>
        <listprice>24.99</listprice>
       <price>17.55</price>
       <edition>4</edition>
       <publisher>Prentice Hall</publisher>
    </book>
</books>
```

Now create a XSL file and write appropriate XSLT rules so that a XSLT processor generate following HTML output (look carefully it only shows books those are available)

**Available books in store:**

| Name | Author | Price | Edition | Publisher |
|---|---|---|---|---|
| SQL ins and outs | Clerk | $17.55 | 4 | QUE |
| OOSAD using UML | Bennet | $17.55 | 3 | Prentice Hall |
| XML | Bennet | $17.55 | 2 | Prentice Hall |