1. Do the workshop in the chapter 3 & 4
2. After finishing the workshop students must complete the following tasks:
   Document Type Definitions
   - Internal DTD
   - External DTD

   XML Schema

3. Document Type Definitions:

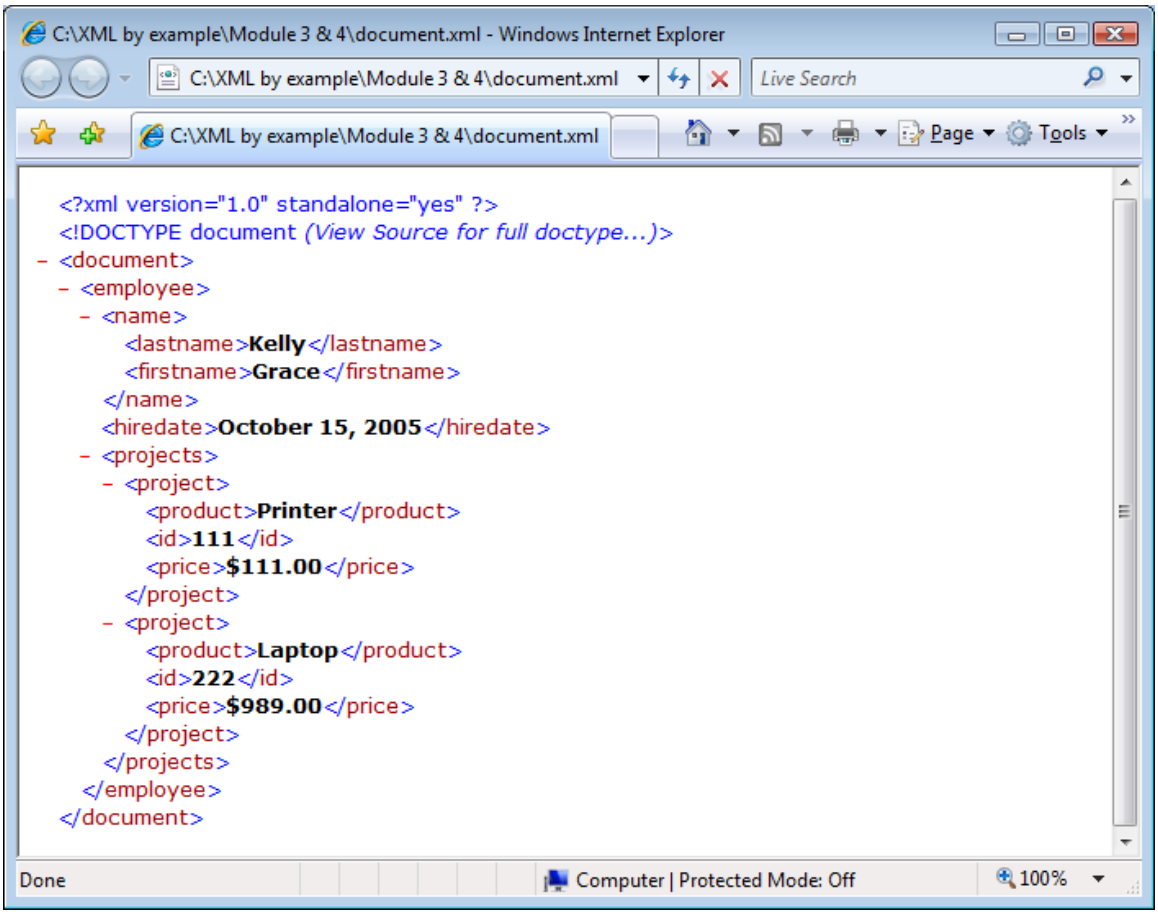
a. Internal DTD

Open the editor and type the following code:

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
<!ELEMENT document (employee)*>
<!ELEMENT employee (name, hiredate, projects)>
<!ELEMENT name (lastname, firstname)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product,id,price)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<document>
    <employee>
        <name>
            <lastname>Kelly</lastname>
            <firstname>Grace</firstname>
        </name>
        <hiredate>October 15, 2005</hiredate>
        <projects>
            <project>
                <product>Printer</product>
                <id>111</id>
                <price>$111.00</price>
            </project>
            <project>
                <product>Laptop</product>
                <id>222</id>
                <price>$989.00</price>
            </project>
        </projects>
    </employee>
</document>
```

Save the code as **document.xml** file and open it in the browser

The output:



You can use the +, *, and ? symbols in content model sequences. For example, here's how you might specify that there can be one or more <name> elements for an employee, an optional <hiredate> element, and any number of <project> elements.

Like this:

```
<?xml version = "1.0" standalone="yes"?>
<!DOCTYPE document [
<!ELEMENT document (employee)*>
<!ELEMENT employee (name+, hiredate?, projects*)>
<!ELEMENT name (lastname, firstname)>
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT hiredate (#PCDATA)>
<!ELEMENT projects (project)*>
<!ELEMENT project (product,id,price)>
<!ELEMENT product (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<document>
    <employee>
        <name>
            <lastname>Kelly</lastname>
```

```
            <firstname>Grace</firstname>
        </name>
        <hiredate>October 15, 2005</hiredate>
        <projects>
            <project>
                <product>Printer</product>
                <id>111</id>
                <price>$111.00</price>
            </project>
            <project>
                <product>Laptop</product>
                <id>222</id>
                <price>$989.00</price>
            </project>

        </projects>
    </employee>
</document>
```

Save and open again in the browser

b. External DTD:

Type and save the following codes as in turn **Book.xml** and **Book.dtd** files:

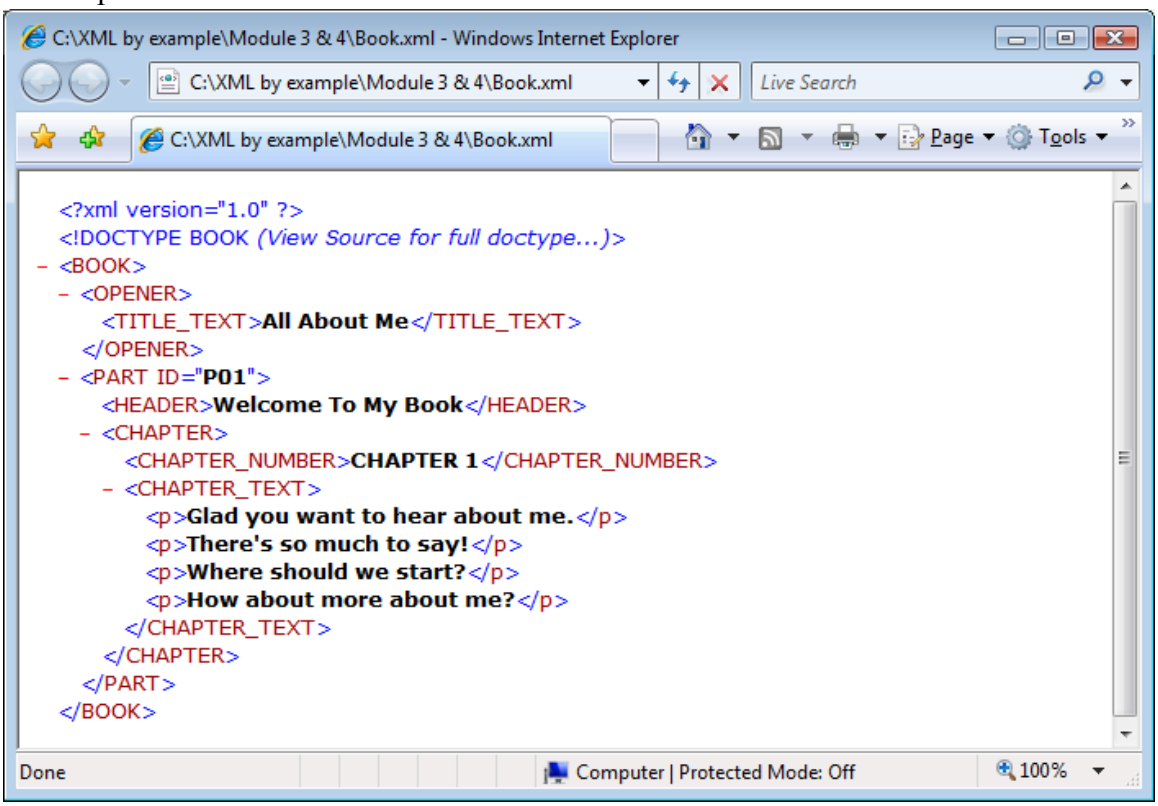
```
<?xml version="1.0"?>
<!DOCTYPE BOOK System ”Book.dtd”>
<BOOK>
    <OPENER>
        <TITLE_TEXT>
            All About Me
        </TITLE_TEXT>
    </OPENER>
    <PART ID=”P01”>
        <HEADER>Welcome To My Book</HEADER>
        <CHAPTER>
            <CHAPTER_NUMBER>CHAPTER 1</CHAPTER_NUMBER>
            <CHAPTER_TEXT>
                <p>&GLAD;</p>
                <p>There's so much to say!</p>
                <p>Where should we start?</p>
                <p>How about more about me?</p>
            </CHAPTER_TEXT>
        </CHAPTER>
    </PART>
</BOOK>
```

```
<!ELEMENT p (#PCDATA)>
    <!ELEMENT BOOK          (OPENER,SUBTITLE?,INTRODUCTION?,(SECTION |
PART)+)>
    <!ELEMENT OPENER        (TITLE_TEXT)*>
    <!ELEMENT TITLE_TEXT    (#PCDATA)>
```

```
    <!ELEMENT SUBTITLE      (#PCDATA)>
    <!ELEMENT INTRODUCTION (HEADER, p+)+>
    <!ELEMENT PART          (HEADER, CHAPTER+)>
    <!ELEMENT SECTION       (HEADER, p+)>
    <!ELEMENT HEADER        (#PCDATA)>
    <!ELEMENT CHAPTER       (CHAPTER_NUMBER, CHAPTER_TEXT)>
    <!ELEMENT CHAPTER_NUMBER (#PCDATA)>
    <!ELEMENT CHAPTER_TEXT (p)+>
    <!ATTLIST PART ID CDATA #REQUIRED>
    <!ENTITY GLAD "Glad you want to hear about me.">
```

Open **Book.xml** file in the browser

The output:



4. XML Schema:

Type and save the following codes as in turn **CustomerDocument.xml** and **CustomerDocument.xsd** files:

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns="http://xmlpowercorp"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlpowercorp CustomerDocument.xsd"
documentDate="2005-03-02">
```

```
    <comment>Good risk</comment>
    <mortgagee phone="888.555.1234">
        <name>James Blandings</name>
        <location>1234 299th St</location>
        <city>New York</city>
        <state>NY</state>
    </mortgagee>
    <mortgages>
        <mortgage loanNumber="66 7777 88">
            <property>The Hackett Place</property>
            <date>2005-03-01</date>
            <loanAmount>80000</loanAmount>
            <term>15</term>
        </mortgage>
        <mortgage loanNumber="11 8888 22">
            <property>123 Acorn Drive</property>
            <date>2005-03-01</date>
            <loanAmount>90000</loanAmount>
            <term>15</term>
        </mortgage>
        <mortgage loanNumber="33 4444 11">
            <property>99 West Pocusset St</property>
            <date>2005-03-02</date>
            <loanAmount>100000</loanAmount>
            <term>30</term>
        </mortgage>
        <mortgage loanNumber="55 3333 88">
            <property>19 Johnson Place</property>
            <date>2005-03-02</date>
            <loanAmount>110000</loanAmount>
            <term>30</term>
        </mortgage>
        <mortgage loanNumber="22 6666 99">
            <property>345 Notingham Court</property>
            <date>2005-03-02</date>
            <loanAmount>120000</loanAmount>
            <term>30</term>
        </mortgage>

    </mortgages>
    <bank phone="888.555.8888">
        <name>XML Bank</name>
        <location>12 Schema Place</location>
        <city>New York</city>
        <state>NY</state>
    </bank>
</document>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://xmlpowercorp"
xmlns="http://xmlpowercorp"
```

```
elementFormDefault="qualified">
    <xsd:annotation>
        <xsd:documentation>
            Mortgage record XML schema.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="document" type="documentType"/>
    <xsd:complexType name="documentType">
        <xsd:sequence>
            <xsd:element ref="comment" minOccurs="1"/>
            <xsd:element name="mortgagee" type="recordType"/>
            <xsd:element name="mortgages" type="mortgagesType"/>
            <xsd:element name="bank" type="recordType"/>
        </xsd:sequence>
        <xsd:attribute name="documentDate" type="xsd:date"/>
    </xsd:complexType>
    <xsd:complexType name="recordType">
        <xsd:sequence>
            <xsd:element name="name" type="xsd:string"/>
            <xsd:element name="location" type="xsd:string"/>
            <xsd:element name="city" type="xsd:string"/>
            <xsd:element name="state" type="xsd:string"/>

        </xsd:sequence>
        <xsd:attribute name="phone" type="xsd:string"
            use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="mortgagesType">
        <xsd:sequence>
            <xsd:element name="mortgage" minOccurs="1" maxOccurs="8">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="property" type="xsd:string"/>
                        <xsd:element name="date" type="xsd:date"
                            minOccurs="0"/>

                        <xsd:element name="loanAmount"
type="xsd:decimal"/>
                        <xsd:element name="term">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:integer">
                                <xsd:maxInclusive value="30"/>
                            </xsd:restriction>
                        </xsd:simpleType>
                        </xsd:element>
                    </xsd:sequence>
                    <xsd:attribute name="loanNumber"
type="loanNumberType"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:simpleType name="loanNumberType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="\d{2} \d{4} \d{2}"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:element name="comment" type="xsd:string"/>
</xsd:schema>
```

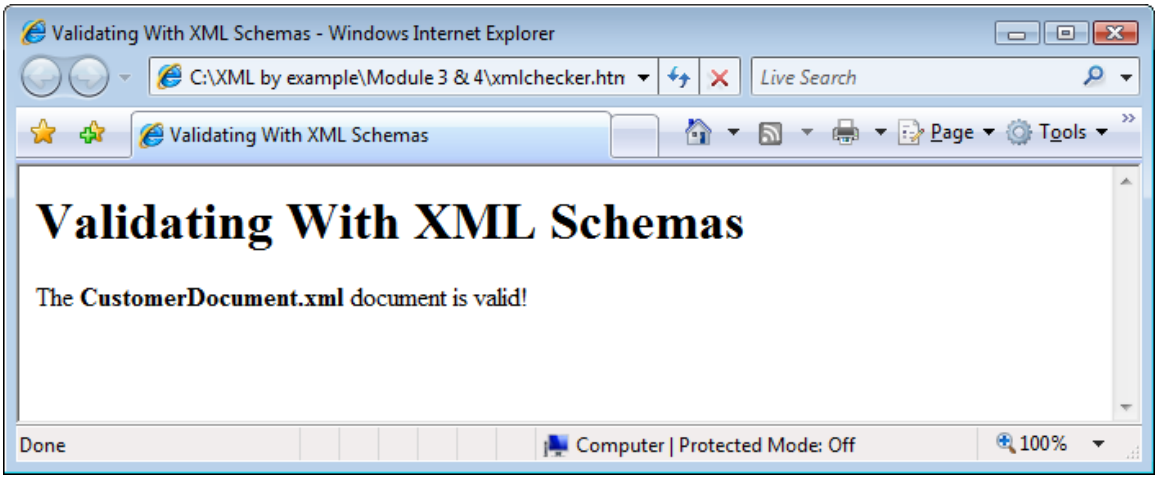
Use XML validators or the code below to valid the **CustomerDocument.xml** file

```
<HTML>
    <HEAD>
        <TITLE>
            Validating With XML Schemas
        </TITLE>
        <SCRIPT LANGUAGE="JavaScript">
            document.write("<H1>Validating With XML
Schemas</H1>");
            var parser = new
ActiveXObject("MSXML2.DOMDocument.4.0");
            parser.validateOnParse = true;

            if (parser.load("CustomerDocument.xml")) {
                document.write("The <b>CustomerDocument.xml<b>
document is valid!");
            } else {
                if (parser.parseError.errorCode != 0) {
                    document.write(parser.parseError.reason);
                }
            }
        </SCRIPT>
    </HEAD>

    <BODY></BODY>
</HTML>
```

The output:



Do It Yourself

2.1. Create a new XML document that uses a DTD to declare an optional `CDATA` attribute named `date` that holds dates in the form 4/1/05, an attribute called `sex` that can take the values `"male"` and `"female"` only, and a required `CDATA name` attribute. Test your work by using an online XML validator.

2.2. Create an XML schema for an XML document that uses the namespace `http://xml21`, with the document element `<document>` and containing both a `<movieTitle>` (content type: `xsd:string`) and `<movieLength>` (content type: `xsd:int`) element.

2.3. Modify the XML document you created in Exercise 2 so that the `<movieTitle>` and `<movieLength>` elements can support `date` attributes of the `xsd:date` type.

2.4. Create an XML document that keeps track of the amount of money owed to you by various friends. In the corresponding XML schema, use the appropriate facets to be sure the amount owed is greater than 0 and less than 5,000 (or 500,000—if you really trust your friends)

2.5. Add another element—the `<repayment>` element—to the XML document you created in Exercise 4. Use the `enumeration` facet to ensure that this element can only take the text values `"Monday"`, `"Tuesday"`, `"Wednesday"`, `"Thursday"`, and `"Friday"`.