

```
1: <?xml version="1.0"?>
2: <BookStore xmlns="http://www.books.org"
3:           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4:           xsi:schemaLocation=
5:             "http://www.books.org
6:             BookStore.xsd">
7:   <Book>
8:     <Title>My Life and Times</Title>
9:     <Author>Paul McCartney</Author>
10:    <Date>1998</Date>
11:    <ISBN>1-56592-235-2</ISBN>
12:    <Publisher>McMillin Publishing</Publisher>
13:  </Book>
14:  <Book>
15:    <Title>Illusions The Adventures of a Reluctant Messiah</Title>
16:    <Author>Richard Bach</Author>
17:    <Date>1977</Date>
18:    <ISBN>0-440-34319-4</ISBN>
19:    <Publisher>Dell Publishing Co.</Publisher>
20:  </Book>
21:  <Book>
22:    <Title>The First and Last Freedom</Title>
23:    <Author>J. Krishnamurti</Author>
24:    <Date>1954</Date>
25:    <ISBN>0-06-064831-7</ISBN>
26:    <Publisher>Harper & Row</Publisher>
27:  </Book>
28: </BookStore>
```

```
1: <?xml version="1.0" encoding="UTF-8"?>
2: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3:           targetNamespace="http://www.books.org"
4:           xmlns="http://www.books.org"
5:           elementFormDefault="qualified">
6:   <xsd:element name="BookStore">
7:     <xsd:complexType>
8:       <xsd:sequence>
9:         <xsd:element ref="Book" minOccurs="1" maxOccurs="unbounded"/>
10:      </xsd:sequence>
11:    </xsd:complexType>
12:  </xsd:element>
13:  <xsd:element name="Book">
14:    <xsd:complexType>
15:      <xsd:sequence>
16:        <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
17:        <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
18:        <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
19:        <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
20:        <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
21:      </xsd:sequence>
22:    </xsd:complexType>
23:  </xsd:element>
24:  <xsd:element name="Title" type="xsd:string"/>
25:  <xsd:element name="Author" type="xsd:string"/>
26:  <xsd:element name="Date" type="xsd:string"/>
27:  <xsd:element name="ISBN" type="xsd:string"/>
28:  <xsd:element name="Publisher" type="xsd:string"/>
29: </xsd:schema>
30:
```

```
1: <?xml version="1.0"?>
2: <schema xmlns="http://www.w3.org/2001/XMLSchema"
3:         targetNamespace="http://www.books.org"
4:         xmlns:bk="http://www.books.org"
5:         elementFormDefault="qualified">
6:   <element name="BookStore">
7:     <complexType>
8:       <sequence>
9:         <element ref="bk:Book" minOccurs="1" maxOccurs="unbounded"/>
10:      </sequence>
11:    </complexType>
12:  </element>
13:  <element name="Book">
14:    <complexType>
15:      <sequence>
16:        <element ref="bk:Title"/>
17:        <element ref="bk:Author"/>
18:        <element ref="bk:Date"/>
19:        <element ref="bk:ISBN"/>
20:        <element ref="bk:Publisher"/>
21:      </sequence>
22:    </complexType>
23:  </element>
24:  <element name="Title" type="string"/>
25:  <element name="Author" type="string"/>
26:  <element name="Date" type="string"/>
27:  <element name="ISBN" type="string"/>
28:  <element name="Publisher" type="string"/>
29: </schema>
```

```
1: <?xml version="1.0"?>
2: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3:           targetNamespace="http://www.books.org"
4:           xmlns="http://www.books.org"
5:           elementFormDefault="qualified">
6:   <xsd:element name="BookStore">
7:     <xsd:complexType>
8:       <xsd:sequence>
9:         <xsd:element name="Book" maxOccurs="unbounded">
10:          <xsd:complexType>
11:            <xsd:sequence>
12:              <xsd:element name="Title" type="xsd:string"/>
13:              <xsd:element name="Author" type="xsd:string"/>
14:              <xsd:element name="Date" type="xsd:string"/>
15:              <xsd:element name="ISBN" type="xsd:string"/>
16:              <xsd:element name="Publisher" type="xsd:string"/>
17:            </xsd:sequence>
18:          </xsd:complexType>
19:        </xsd:element>
20:      </xsd:sequence>
21:    </xsd:complexType>
22:  </xsd:element>
23: </xsd:schema>
```

```
1: <?xml version="1.0"?>
2: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3:           targetNamespace="http://www.books.org"
4:           xmlns="http://www.books.org"
5:           elementFormDefault="qualified">
6:   <xsd:element name="BookStore">
7:     <xsd:complexType>
8:       <xsd:sequence>
9:         <xsd:element name="Book" type="BookPublication" maxOccurs="unbounded"/>
10:      </xsd:sequence>
11:    </xsd:complexType>
12:  </xsd:element>
13:  <xsd:complexType name="BookPublication">
14:    <xsd:sequence>
15:      <xsd:element name="Title" type="xsd:string"/>
16:      <xsd:element name="Author" type="xsd:string"/>
17:      <xsd:element name="Date" type="xsd:string"/>
18:      <xsd:element name="ISBN" type="xsd:string"/>
19:      <xsd:element name="Publisher" type="xsd:string"/>
20:    </xsd:sequence>
21:  </xsd:complexType>
22: </xsd:schema>
```

```
1: <?xml version="1.0"?>
2: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3:           targetNamespace="http://www.books.org"
4:           xmlns="http://www.books.org"
5:           elementFormDefault="qualified">
6:   <xsd:simpleType name="ISBN-type">
7:     <xsd:restriction base="xsd:string">
8:       <xsd:pattern value="\d{1}-\d{5}-\d{3}-\d{1}|\d{1}-\d{3}-\d{5}-\d{1}|\d{1}-\d{2}-\d{6}-\d{1}"/>
9:     </xsd:restriction>
10:  </xsd:simpleType>
11:  <xsd:element name="BookStore">
12:    <xsd:complexType>
13:      <xsd:sequence>
14:        <xsd:element name="Book" maxOccurs="unbounded">
15:          <xsd:complexType>
16:            <xsd:sequence>
17:              <xsd:element name="Title" type="xsd:string"/>
18:              <xsd:element name="Author" type="xsd:string"/>
19:              <xsd:element name="Date" type="xsd:gYear"/>
20:              <xsd:element name="ISBN" type="ISBN-type"/>
21:              <xsd:element name="Publisher" type="xsd:string"/>
22:            </xsd:sequence>
23:          </xsd:complexType>
24:        </xsd:element>
25:      </xsd:sequence>
26:    </xsd:complexType>
27:  </xsd:element>
28: </xsd:schema>
```

```
1: <?xml version="1.0"?>
2: <purchaseOrder orderDate="1999-10-20"
3:     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4:     xsi:noNamespaceSchemaLocation="po.xsd">
5:   <shipTo country="US">
6:     <name>Alice Smith</name>
7:     <street>123 Maple Street</street>
8:     <city>Mill Valley</city>
9:     <state>CA</state>
10:    <zip>90952</zip>
11:  </shipTo>
12:  <billTo country="US">
13:    <name>Robert Smith</name>
14:    <street>8 Oak Avenue</street>
15:    <city>Old Town</city>
16:    <state>PA</state>
17:    <zip>95819</zip>
18:  </billTo>
19:  <comment>Hurry, my lawn is going wild!</comment>
20:  <items>
21:    <item partNum="872-AA">
22:      <productName>Lawnmower</productName>
23:      <quantity>1</quantity>
24:      <USPrice>148.95</USPrice>
25:      <comment>Confirm this is electric</comment>
26:    </item>
27:    <item partNum="926-AA">
28:      <productName>Baby Monitor</productName>
29:      <quantity>1</quantity>
30:      <USPrice>39.98</USPrice>
31:      <shipDate>1999-05-21</shipDate>
32:    </item>
33:  </items>
34: </purchaseOrder>
```

```
1: <?xml version="1.0"?>
2:
3: <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
4:
5:   <xsd:annotation>
6:     <xsd:documentation xml:lang="en">
7:       Purchase order schema.
8:       Copyright 2007 Harvard University. All rights reserved.
9:     </xsd:documentation>
10:  </xsd:annotation>
11:
12:  <xsd:element name="purchaseOrder" type="PurchaseOrderType"/>
13:
14:  <xsd:element name="comment" type="xsd:string"/>
15:
16:  <xsd:complexType name="PurchaseOrderType">
17:    <xsd:sequence>
18:      <xsd:element name="shipTo" type="USAddress"/>
19:      <xsd:element name="billTo" type="USAddress"/>
20:      <xsd:element ref="comment" minOccurs="0"/>
21:      <xsd:element name="items" type="Items"/>
22:    </xsd:sequence>
23:    <xsd:attribute name="orderDate" type="xsd:date"/>
24:  </xsd:complexType>
25:
26:  <xsd:complexType name="USAddress">
27:    <xsd:sequence>
28:      <xsd:element name="name" type="xsd:string"/>
29:      <xsd:element name="street" type="xsd:string"/>
30:      <xsd:element name="city" type="xsd:string"/>
31:      <xsd:element name="state" type="xsd:string"/>
32:      <xsd:element name="zip" type="xsd:decimal"/>
33:    </xsd:sequence>
34:    <xsd:attribute name="country" type="xsd:NMTOKEN" fixed="US"/>
35:  </xsd:complexType>
36:
37:  <xsd:complexType name="Items">
38:    <xsd:sequence>
39:      <xsd:element name="item" minOccurs="0" maxOccurs="unbounded">
40:        <xsd:complexType>
41:          <xsd:sequence>
42:            <xsd:element name="productName" type="xsd:string"/>
43:            <xsd:element name="quantity">
44:              <xsd:simpleType>
45:                <xsd:restriction base="xsd:positiveInteger">
```



```
46:         <xsd:maxExclusive value="100"/>
47:         </xsd:restriction>
48:     </xsd:simpleType>
49: </xsd:element>
50:     <xsd:element name="USPrice" type="xsd:decimal"/>
51:     <xsd:element ref="comment" minOccurs="0"/>
52:     <xsd:element name="shipDate" type="xsd:date" minOccurs="0"/>
53: </xsd:sequence>
54:     <xsd:attribute name="partNum" type="SKU" use="required"/>
55: </xsd:complexType>
56: </xsd:element>
57: </xsd:sequence>
58: </xsd:complexType>
59:
60: <!-- Stock Keeping Unit, a code for identifying products -->
61: <xsd:simpleType name="SKU">
62:     <xsd:restriction base="xsd:string">
63:         <xsd:pattern value="\d{3}-[A-Z]{2}"/>
64:     </xsd:restriction>
65: </xsd:simpleType>
66:
67: </xsd:schema>
```

```
1: import javax.xml.parsers.SAXParser;
2: import javax.xml.parsers.SAXParserFactory;
3:
4: import org.xml.sax.EntityResolver;
5: import org.xml.sax.InputSource;
6: import org.xml.sax.SAXParseException;
7:
8: import org.xml.sax.helpers.DefaultHandler;
9:
10:
11: /**
12:  * Lecture 9's demonstration of validation
13:  * by XML DTD or XML Schema.
14:  *
15:  * @author Computer Science E-259
16:  */
17:
18: public class SAXValidator2 extends DefaultHandler
19: {
20:     /**
21:      * Main driver. Expects one command-line argument:
22:      * the name of the XML file to validate
23:      *
24:      * @param argv [0] - filename
25:      */
26:     public static void main(String [] argv)
27:     {
28:         if (argv.length == 0)
29:         {
30:             System.out.println("Usage: SAXValidator2 file [dtd|xsd]");
31:             System.exit(1);
32:         }
33:
34:         // grab filename
35:         String input = argv[0];
36:
37:         // grab validation mechanism, if any
38:         String validator = (argv.length > 1) ? argv[1] : null;
39:
40:         // process input as requested
41:         try
42:         {
43:             // instantiate a reference to a SAX parser
44:             SAXParser parser = null;
45:
```

```
46:         // instantiate a SAX parser factory
47:         SAXParserFactory factory = SAXParserFactory.newInstance();
48:
49:         // instantiate a SAX parser, enabling validation as requested
50:         if (validator != null && validator.equals("dtd"))
51:         {
52:             factory.setValidating(true);
53:             parser = factory.newSAXParser();
54:             System.out.println("Validation by DTD on.");
55:         }
56:         else if (validator != null && validator.equals("xsd"))
57:         {
58:             factory.setNamespaceAware(true);
59:             factory.setValidating(true);
60:             parser = factory.newSAXParser();
61:             parser.setProperty
62:             (
63:                 "http://java.sun.com/xml/jaxp/properties/schemaLanguage",
64:                 "http://www.w3.org/2001/XMLSchema"
65:             );
66:             System.out.println("Validation by XML Schema on.");
67:         }
68:         else
69:         {
70:             factory.setValidating(false);
71:             parser = factory.newSAXParser();
72:             System.out.println("Validation off.");
73:         }
74:
75:         // instantiate our little handler
76:         SAXValidator2 handler = new SAXValidator2();
77:
78:         // parse the file
79:         parser.parse(input, handler);
80:
81:     } catch (Exception e) {
82:         e.printStackTrace();
83:     }
84: }
85:
86:
87: /**
88:  * Receive notification of a recoverable parser error.
89:  *
90:  * @param e the exception thrown
```

```
91:     */
92:     public void error (SAXParseException e)
93:     {
94:         System.out.println("Parsing error:  " + e.getMessage());
95:     }
96:
97:
98:     /**
99:     * Receive notification of a parser warning.
100:    *
101:    * @param e  the exception thrown
102:    */
103:    public void warning (SAXParseException e)
104:    {
105:        System.out.println("Parsing warning:  " + e.getMessage());
106:    }
107:
108:
109:    /**
110:    * Report a fatal XML parsing error.
111:    *
112:    * @param e  the exception thrown
113:    */
114:    public void fatalError (SAXParseException e)
115:    {
116:        System.out.println("Fatal parsing error:  " + e.getMessage());
117:        System.exit(1);
118:    }
119: }
```