

Matric No: _____

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

CO32036

INTERNET MARK-UP LANGUAGES

ACADEMIC SESSION: 2005-2006

EXAM DIET: AUGUST

TRIMESTER: ONE

EXAM DURATION: 2 HOURS

READING TIME: NONE

EXAM PAPER INFORMATION

Answer any THREE questions.

Number of questions – FIVE

Number of pages – EIGHT

Number of sections – ONE

EXAMINER: ANDREW CUMMING

PLEASE READ FULL INSTRUCTIONS BEFORE COMMENCING WRITING

1. (a) Explain how one can check for validity in an XML document. Give an example of an application where validation would be important. (5)

- (b) The following statements have been taken from the W3 definition of XML 1.0:

```
Comment ::= '
```

2. (a) In the following Document Type Definition (DTD), state in words the permitted combinations of children for the element 'country'

```
<!ELEMENT country (name,(president,monarch))>
<!ELEMENT name      (#PCDATA)>
<!ELEMENT president (#PCDATA)>
<!ELEMENT monarch   (#PCDATA)>
```

(7)

- (b) The ELEMENT definition for 'country' is then changed. It should instead obey the following rules:

- Every country must have a name as the first node.
- Every country must have a president or a monarch, it may not have both.

Give a corrected version for the definition of 'country'.

(7)

- (c) Distinguish between the "narrative oriented" and the "data oriented" styles of XML format. Illustrate your answer using examples drawn from XHTML and SVG. You may find the following code fragments useful.

(11)

XHTML example, from www.w3.org:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="en-US" lang="en-US">
<head profile="http://www.w3.org/2000/08/w3c-synd/#">
<title>World Wide Web Consortium</title>
</head>
<body>
<h1 id="logo">
    </h1>
<h2 id="slogan">Leading the Web to Its Full
Potential...</h2>
</body>
</html>
```

SVG Example, the Flag of Panama:

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "../../../svg11-flat.dtd">
<svg width="12cm" height="8cm" viewBox="0 0 12 8">
    <path fill="blue" transform="translate(3,2)"
        d="M 0 -1 L 0.588 0.809 -0.951 -0.309
            0.951 -0.309 -0.588 0.809 Z"/>
    <path fill="red" transform="translate(9,6)"
        d="M 0 -1 L 0.588 0.809 -0.951 -0.309
            0.951 -0.309 -0.588 0.809 Z"/>
    <rect x="0" y="4" width="6" height="4" fill="blue"/>
    <rect x="6" y="0" width="6" height="4" fill="red"/>
</svg>
```

Total marks [25]

3. (a) Distinguish between well-formed and valid document. (5)
- (b) Consider the following XSD Schema document.

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="country">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="capital" type="xs:string"/>
<xs:element name="king" type="xs:string"
minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="queen" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

- (i) State in words the permitted combinations of children for the element 'country'. (5)
- (ii) Give an illustrative example of a valid document. (5)
- (iii) Give an illustrative example of an invalid document. (5)
- (c) The definition for 'country' is then changed. It should instead obey the following rules:
- Every country must have a name as the first node.
 - Every country must have a capital city as the following node.
 - A country may have a king.
 - A country may have a queen.

Give a corrected version of the definition for 'country'. (5)

Total marks [25]

4.

```
1 import org.w3c.dom.*;
2 import javax.xml.parsers.DocumentBuilder;
3 import javax.xml.parsers.DocumentBuilderFactory;
4
5 public class D
6 {
7     public static void main(String[] args) throws Exception
8     {
9         Document d =
10             DocumentBuilderFactory.newInstance()
11                 .newDocumentBuilder()
12                 .parse("../cia.xml");
13         Element e = (Element)d.getDocumentElement().getFirstChild();
14         while (e!=null)
15         {
16             if (e.getAttribute("name").equals("Albania"))
17                 System.out.println(e.getNodeName());
18             e = (Element)e.getNextSibling();
19         }
20     }
21 }
```

```
<cia>
  <country name='Afghanistan' region='Asia' area='652000'
    population='25838797' gdp='21000000000'/>
  <country name='Albania' region='Europe' area='28748'
    population='3490435' gdp='5600000000'/>
  <country name='Algeria' region='Africa' area='2381740'
    population='31193917' gdp='147600000000'/>
</cia>
```

Consider the DOM program D.java shown. It is executed with the XML file cia.xml.

- (a) State the output of the program. (6)
- (b) With a different input file it is possible for the cast on line 18 to fail at run time. Describe an input file that could cause such an error. (6)
- (c) We might have chosen to declare e as a Node rather than as an Element. Two casts would no longer be required. One new cast would be required. Describe the new required cast. (6)
- (d) The W3C recommendation for DOM concedes that "DOM names tend to be long and descriptive". Give two reasons for this decision. (7)

Total marks [25]

While it would be nice to have attribute and method names that are short, informative, internally consistent, and familiar to users of similar APIs, ... DOM names tend to be long and descriptive ...

5. The SAX program shown on the following page is executed taking the XML file shown as input.

(a) State the output of the program. (4)

- (b) The program is required to calculate the total of the population. Indicate how the program should be changed to meet this requirement. You may find the following phrases a useful reminder of Java syntax.

```
//String to long conversion in Java
long n = Long.parseLong("1234");
//SAX interface for Attributes includes the method getValue
String s = atts.getValue("region");
```

(9)

(c) State why this file includes the "Do nothing methods". (4)

(d) State the circumstances that might generate a SAXException. (4)

(e) The function performed by this program could be completed much more easily using the DOM API. (4)

Total marks [25]

NodeCounter.java

```
1 import org.xml.sax.*;
2
3 public class NodeCounter implements ContentHandler {
4
5     int nodeCount;
6
7     public void startDocument() throws SAXException
8     {
9         nodeCount=0;
10    }
11
12    public void startElement(String namespaceURI, String localName,
13        String qualifiedName, Attributes atts) throws SAXException
14    {
15        nodeCount++;
16    }
17
18    public void endDocument() throws SAXException
19    {
20        System.out.println("Number of nodes: " + nodeCount);
21    }
22
23    // Do-nothing methods
24    public void setDocumentLocator(Locator l) {}
25    public void startPrefixMapping(String p, String u)
26        throws SAXException {}
27    public void endPrefixMapping(String p) throws SAXException {}
28    public void skippedEntity(String n) throws SAXException {}
29    public void processingInstruction(String t, String d)
30        throws SAXException {}
31    public void characters(char[] t, int s, int l)
32        throws SAXException {}
33    public void ignorableWhitespace(char[] t, int s, int l)
34        throws SAXException {}
35    public void endElement(String namespaceURI, String localName,
36        String qualifiedName) throws SAXException {}
37 }
```

cia.xml

```
<cia>
  <country name='Afghanistan' region='Asia' area='652000'
    population='25838797' gdp='210000000000' />
  <country name='Albania' region='Europe' area='28748'
    population='3490435' gdp='56000000000' />
  <country name='Algeria' region='Africa' area='2381740'
    population='31193917' gdp='1476000000000' />
</cia>
```

End of Paper