

Instructions to Examinees:

- This is a close book and close note exam.
- Honesty policy is strictly enforced. You could get zero because of dishonesty.
- Please show all work on exam sheets. You can write on the backside of the exam sheets.
- Time is limited. Please write the key points.
- Any assumption made in answering questions should be stated.
- Points for this paper total 200.

Question 1 - Web Technologies (21 points)

What do these terminologies stand for?

1. DHTML - Dynamic HyperText Markup Language
2. DOM - Document Object Model
3. SAX - Simple API for XML
4. SGML - Standard Generalized Markup Language
5. SOAP - Simple Object Access Protocol
6. XML - Extensible Markup Language
7. XSL - Extensible Stylesheet Language

Question 2 - Perl and CGI (16 points)

The following URL is typical of those observed when performing queries to a search engine:

`http://search.yahoo.com/bin/search?p=%22electronic+commerce%22`

1. What is the difference between the GET and POST methods?
2. Can you tell which HTML form method is used from this URL? Give the reason for your answer.
3. What is the name of the CGI program which is executed by the Web server? What is the user input from the HTML form?
4. How does the CGI program access the data which is submitted when this URL is accessed?

Ans: 1. The difference is how the data is retrieved by the CGI program.
In GET method CGI data is formatted in two pieces, separated by a ?
The formatted user input is placed in the environment variable
QUERY_STRING, which is limited to 1024 characters. POST directs the
formatted user input into the standard input buffer.
2. It was specified as GET because the query string is separated by a ?.
3. search is the CGI program.
4. The data is accessed through the environment variable QUERY_STRING.

Question 3 - Internet Database (8 points)

What are three components you need to provide online database access?

Ans: The three components are:
(1) A database that can be created in any computer.
(2) A script or program to reside on the server that can access the data.
(3) A CGI form or applet to send queries to the script or program on the server.

Question 4 - Security and E-Commerce (25 points)

1. What is PGP? (5 points)
2. What is TLS? (5 points)
3. What is SET? (5 points)
4. What is a digital certificate? What information is contained in a digital certificate? (10 points)

Ans: 1. Pretty Good Privacy (PGP) is a popular program used to encrypt and decrypt e-mail over the Internet.

2. TLS (Transport Layer Security) is a protocol that ensures privacy between communicating applications and their users on the Internet.

3. SET (Secure Electronic Transaction) is a system for ensuring the security of financial transactions on the Internet.

4. A digital certificate is an electronic "credit card" that establishes your credentials when doing business or other transactions on the Web. A digital certificate contains the following six pieces of information:

- (1) The owner's public key
- (2) The owner's identity
- (3) The issuer's identity
- (4) A serial number
- (5) The issuing and expiration dates
- (6) The issuer's digital signature

Question 5 - XML Concepts (35 points)

1. How can XML be used? (10 points)
2. What are three parts of XSL? (10 points)
3. What is a well-formed XML document? (15 points)

Ans: 1. (1) XML can separate Data from HTML.
(2) XML is used to exchange, store, and share data.
(3) XML can make your Data more Useful.
(4) XML can be used to Create new Languages.

2. (1) A method for transforming XML documents
(2) A method for defining XML parts and patterns
(3) A method for formatting XML documents

3. (1) All XML elements must have a closing tag.
(2) XML tags are case sensitive.
(3) All XML elements must be properly nested.
(4) All XML documents must have a root tag.
(5) Attribute values must always be quoted.
(6) Element names must follow naming rules.

Question 6 - Semester Project (20 points)

1. Outline your semester project. (8 points)
2. Specify its main features, current shortcomings, and future development. (12 points)

Question 7 - XML Practice (75 points)

This is some information for your reference:

- Markups for XML, DTD, XSL, and XSD:

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "InternalNote.dtd">
<!DOCTYPE note [
  <!ELEMENT payment (#PCDATA)>
  <!ATTLIST payment type CDATA "check">
]>
<?xml-stylesheet type="text/xsl" href="simple.xsl" ?>
<flashcards xmlns="http://www.net-language.com"
  xsi:noNamespaceSchemaLocation="customer.xsd">
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<xsl:for-each select="classmates/classmate">
<xsl:value-of select="lastname"/>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="publist">
<xs:attribute name="itemtype" type="xs:string"/>
<xs:complexType>
<xs:sequence>
<xml id="cdcat" src="cd_catalog.xml"></xml>
<table border="1" datasrc="#cdcat">
<span datafld = "LastName">
```

- Most common types of XML Schema are string, decimal, string, decimal, integer, boolean, date, and time.

Based on the last digital of your student ID number, choose the object from the following ones to specify:

Last digital of your ID	Objects
0, 1	Speech Presentations
2, 3	Basketball Games
4, 5	Concerts
6, 7	TV Programs
8, 9	Meeting Schedules

For example, if the last digital of your student ID number is 5, then Concerts is the object you need to specify.

1. Use SQL to create a database table to specify this object. (10 points)
2. Create an XML document to specify this object. (10 points)
3. Create a DTD document for this XML. (10 points)
4. Create an XSL document for this XML. (15 points)
5. Create an XML Schema document for this XML. (15 points)
6. Create a HTML document to load this XML in the XML Data Island. (15 points)

```
Ans: 1. create table concerts (
  performer varchar[40],
  place varchar[20],
  date date
);
```

2.

```
<?xml version="1.0"?>
<concerts>
  <concert>
    <performer>Celine Dion</performer>
    <place>Wichita Studium</place>
    <date>2003-1-10</date>
  </concert>
  <concert>
    <performer>Paul McCartney</performer>
    <place>Kansas Studium</place>
    <date>2003-1-20</date>
  </concert>
</concerts>
```
3.

```
<!DOCTYPE concerts [
<!ELEMENT concerts (concert)>
<!ELEMENT cconcert (performer, place, date)>
<!ELEMENT performer (#PCDATA)>
<!ELEMENT place (#PCDATA)>
<!ELEMENT date (#PCDATA)>
]>
```
4.

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
    <body>
    <table>
    <tr>
      <th>Performer</th>
      <th>Place</th>
      <th>Date</th>
    </tr>
    <xsl:for-each select="concerts/concert" order-by="+ date">
    <tr>
      <td><xsl:value-of select="performer" /></td>
      <td><xsl:value-of select="place" /></td>
      <td><xsl:value-of select="date" /></td>
    </tr>
    </xsl:for-each>
    </table>
    </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```
5.

```
<?xml version="1.0" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:complexType name="concerts">
  <xs:sequence>
    <xs:complexType name="concert">
      <xs:sequence>
        <xs:element name="performer" type="xs:string"/>
        <xs:element name="place" type="xs:string"/>
        <xs:element name="date" type="xs:date"/>
      </xs:sequence>
    </xs:complexType>
```

```
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```
6. <html>
  <body>
    <xml src="concert.xml" id = "xmlDoc"></xml>
    <table border = "1" datasrc = "#xmlDoc">
      <tr>
        <th>Performer</th>
        <th>Place</th>
        <th>Time</th>
      </tr>
      <tr>
        <td><span datafld = "performer"></span></td>
        <td><span datafld = "place"></span></td>
        <td><span datafld = "date"></span></td>
      </tr>
    </table>
  </body>
</html>
```