

CS 665 Information Delivery on the Internet Final Exam - Name: \_\_\_\_\_  
Fall 2002

Part 1: (75 points - 3 points for each problem)

- ( A ) 1. What protocol is used by all Web communications transactions?  
(A) HTTP (B) TCP/IP (C) FTP (D) PPP
- ( D ) 2. The default port for the HTTP server is:  
(A) 22 (B) 25 (C) 79 (D) 80
- ( D ) 3. Which one is not a Web server?  
(A) Apache (B) Tomcat (C) IIS (D) lynx
- ( C ) 4. What is the correct HTML for inserting an image?  
(A) `<image src="car.gif" alt = "image of a car">` (B) `<img alt = "image of a car" >car.gif</img>`  
(C) `` (D) `<img href="car.gif alt = "image of a car">`
- ( B ) 5. What HTML tag can be used for making a menu?  
(A) `<menu>` (B) `<select>` (C) `<input type="select">` (D) `<input type="menu">`
- ( B ) 6. How can you make an e-mail link in HTML?  
(A) `<a href="xxx@yyy">` (B) `<a href="mailto:xxx@yyy">`  
(C) `<mail href="xxx@yyy">` (D) `<mail>xxx@yyy<mail>`
- ( C ) 7. Which variable has the value of the system variable, `errno` in Perl?  
(A) `$?` (B) `$_` (C) `#!` (D) `$$`
- ( A ) 8. If the address of an array is stored in `$value`, how do you get the values of this array?  
(A) `@$value` (B) `\$value` (C) `&$value` (D) `$$value`
- ( D ) 9. How is a dollar sign (\$) encoded in a CGI query string? Its decimal ASCII value is 36.  
(A) `@36` (B) `%36` (C) `@24` (D) `%24`
- ( A ) 10. How does a Perl program specify that it needs to access the database module?  
(A) `use DBI;` (B) `access DBI;` (C) `include DBI;` (D) `define DBI;`
- ( B ) 11. Which directive is used to specify the server root for the Apache server?  
(A) `ServerDirectory` (B) `ServerRoot` (C) `DocumentRoot` (D) `ScriptAlias`
- ( D ) 12. Which Session method can be used to specify the lifetime of a Session object?  
(A) `setLifeTime` (B) `setMaxAge` (C) `setTime` (D) none of above
- ( C ) 13. Which is not the life-cycle method of the servlet interface?  
(A) `init` (B) `service` (C) `operation` (D) `destroy`
- ( C ) 14. When we throw a user-defined exception in Java, the exception is a  
(A) class (B) variable (C) object (D) formal parameter
- ( A ) 15. Given a character `ch` in Java, the next character value alphabetically can be obtained by:  
(A) `ch ++` (B) `nextChar(ch)` (C) `((int) ch) ++` (D) `ch.nextChar()`
- ( B ) 16. How do you specify a column in a table must have a value in SQL?  
(A) not empty (B) not null (C) default value (D) have value
- ( C ) 17. When you are using the `insert` command in a SQL statement, which method of the `Statement` class is used for in JDBC?  
(A) `executeInsert()` (B) `executeQuery()` (C) `executeUpdate()` (D) `actInsert()`

- ( A ) 18. Which HTML tag is used to define an internal style sheet?  
 (A) <style> (B) <script> (C) <css> (D) <sheet>
- ( B ) 19. Which CSS tag is designed for a section of document to appear in a different font size or color?  
 (A) <section> (B) <div> (C) <division> (D) <span>
- ( C ) 20. How to make a list that list its items with squares in CSS?  
 (A) type: 2 (B) type: square (C) list-style-type: square (D) list-type: square
- ( C ) 21. Which property in CSS is used to specify that text should flow around the floating element?  
 (A) flow (B) align (C) float (D) margin
- ( D ) 22. How to insert a comment in your CSS file?  
 (A) ' a comment (B) // a comment (C) /\* a comment // (D) /\* a comment \*/
- ( A ) 23. Which sign is used to specify a parameter entity in DTD?  
 (A) % (B) # (C) ! (D) &
- ( C ) 24. Which is not a correct name for an XML element?  
 (A) <age> (B) <NAME> (C) <first name> (D) none of above
- ( B ) 25. When you modify an XML file, what must you do to display the changes?  
 (A) modify the HTML file (B) modify the XSL file (C) modify the XML file (D) modify the DTD file

**Part 2: (125 points)**

1. Briefly explain these terminologies. If they are acronyms, also write what they stand for. (24 points)
  - (a) **MIME** The Multi-purpose Internet Mail Extensions is an extension used to exchange different kinds of data files on the Internet such as audio, video, images, and application programs.
  - (b) **URL** A Uniform Resource Locator is the address of a file (resource) accessible on the Internet.
  - (c) **XHTML** The Extensible Hypertext Markup Language is the set of markup symbols or codes inserted in a document to make it displayable on the World Wide Web. It is an application of the Extensible Markup Language (XML).
  - (d) **CGI** The Common Gateway Interface is the interface between a browser and software on the server.
  - (e) **Proxy Server** A proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the enterprise can ensure security, administrative control, and caching service.
  - (f) **Servlet Engine** A servlet engine is a program that manages servlets which can either be built into the Web server or be added to the Web server.
  - (g) **SGML** Standard Generalized Markup Language is a standard for how to specify a document markup language or tag set.
  - (h) **XSL** Extensible Stylesheet Language, formerly called Extensible Style Language, is a language for creating a style sheet that describes how data sent over the Web using the Extensible Markup Language (XML) is to be presented to the user.
2. Describe three levels of style sheets and their applying range. (6 points)  
 Ans:
  - (a) Inline style sheets apply to the content of a single tag.
  - (b) Document-level style sheets apply to the whole body of a document.
  - (c) External style sheets can apply to the bodies of any number of documents.

3. Give one example for pass-by-value and one for pass-by-reference in Perl? (6 points)  
 Ans:

```

swap(3, 4);
swop(3, 4);

sub swap {          # Call-by-reference
    $temp = $_[0];
    $_[0] = $_[1];
    $_[1] = $temp;
}
sub swop {          # Call-by-value
    my ($a, $b) = @_;
    $temp = $a;
    $a = $b;
    $b = $temp;
}

```

4. Consider the following Perl code (`argv.pl`). Write the results after running it on the command line as `argv.pl 11 22 33 44`. (7 points)

```

#!/usr/bin/perl

print shift, "\n";
push @ARGV, (24, 36, 60);
print pop, "\n";
print $#ARGV, "\n";
unshift @ARGV, 48;
print scalar (@ARGV), "\n";
print pop, "\n";

```

Ans:

```

11
60
4
6
36

```

5. Write a Perl Program with the following specifications: (10 points)

- (a) Input: A text file with the following format:

```

John Wayne, 92 82 88 96 78
Jayne Leno, 87 93 85 82 94
. . . . .

```

Grade	Average
A	90-100
B	80-89
C	70-79
D	60-69
F	0-59

- (b) Output: A list of names, averages, and grades with the grading scale:

Ans:

```

#!/usr/bin/perl

my @grade = ('F', 'D', 'C', 'B', 'A');
while (<>) {
    my ($name, $score) = split(/,/);
    my @scores = split(' ', $score);
    my $sum = 0;
    foreach $score (@scores) {
        $sum += $score;
    }
    $average = $sum/scalar (@scores);
    print "$name $average ", $grade[(($average - 50)/10)], "\n";
}

```

6. Complete the following JDBC servlet which reads a text input form with the attribute name = "price" and replies a list of stocks whose price is larger than \$10 from the Stock table in a MySQL database with specifications as shown in the following two tables: (13 points)

Host name	kira
Account name	stock
Password	stock123
Database	stocks

Field	Type
ticker	varchar(6)
company	varchar(20)
price	decimal(4, 2)

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

public class QueryStock extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException, ClassNotFoundException,
        IllegalAccessException, InstantiationException, SQLException
    {
        Class.forName("com.mysql.jdbc.Driver").newInstance();

        Connection conn = DriverManager.getConnection(
            "jdbc:mysql://kira/stocks?user=stock&password=stock123");

        response.setContentType("text/html");
        PrintWriter returnHTML = response.getWriter();
        int price = Integer.parseInt(request.getParameter("price"));

        returnHTML.println("<html><head><title>Stock</title></head><body>");

        Statement stmt = conn.createStatement();
        String sql_cmd = "select * from stock where price > " + price;
        ResultSet RS = stmt.executeQuery(sql_cmd);

        while (RS.next()) {
            returnHTML.println("Ticker: " + RS.getString(1) + "<br>");
            returnHTML.println("Company: " + RS.getString(2) + "<br>");
            returnHTML.println("Price: " + RS.getFloat(3) + "<br>");
        }

        returnHTML.println("</body></html>");
        returnHTML.close();

        RS.close();
        stmt.close();
        conn.close();
    }
}
```

7. Describe the five most commonly used HTTP request methods. (10 points)

Ans:

GET	Return the contents of the specified document
HEAD	Return the header information for the specified document
POST	Execute the specified document, using the enclosed data
PUT	Replace the specified document with the enclosed data
DELETE	Delete the specified document

8. Describe the XML syntax rules. (12 points)

Ans:

- (a) All XML documents must have a root tag.
  - (b) All XML elements must have a closing tag.
  - (c) All XML elements must be properly nested.
  - (d) XML tags are case sensitive.
  - (e) Attribute values must always be quoted.
  - (f) Element names must follow naming rules.
9. (a) Write a HTML document with a text input form to collect the user's name. This HTML document will request a CGI program `greet.pl` in the `cgi-bin` directory of your account at `kirk.cs.twsu.edu:6555`.
- (b) Write the CGI program `greet.pl` that returns a greeting message with the current date and the user's name.  
Hint: Use the following functions in `CGI.pm`: `param()`, `header()`, `start_html()`, and `end_html()`.

(12 points)

Ans:

- (a) 

```
<html>
<head>
<title>HTML to call the CGI-Perl program greet.pl</title>
</head>
<body>
<form action = "http://kirk.cs.twsu.edu:6555/cgi-bin/cs655/greet.pl"
      method = "POST">
<input type = "text" name = "name" size = "20">
<input type = "submit" value = "Submit">
</form>
</body>
</html>
```
- (b) 

```
#!/usr/bin/perl -w
# greet.pl - a CGI program that returns a greeting to the user

use CGI qw(:standard);

print header();
print start_html("Greeting");
my $name = param("name");
my $date = qx(date);
print "$name! Greetings from your Web server, $date!";
print end_html();
```

10. Based on the last digital of your social security number, choose the object from the following table to specify: (25 points)

Last digital of your SSN	Objects
0, 1	Flight Schedule
2, 3	Theater Schedule
4, 5	Movie DVD
6, 7	Video Game
8, 9	Radio Program

For example, if the last digital of your social security number is 5, then Movie DVD is the object you need to specify.

- Use SQL to create a database table to specify this object.
- Create an XML document to specify this object.
- Create a DTD document for this XML.
- Create an CSS document for this XML.
- Create an XSL document for this XML. (Optional for undergraduate students)

You can use the following information to create XML, DTD, CSS, and XSL documents.

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "InternalNote.dtd">
<!DOCTYPE note [
  <!ELEMENT payment (#PCDATA)>
  <!ATTLIST payment type CDATA "check">
]>
<link rel="stylesheet" type = "text/css" href="simple.css">
<?xml-stylesheet type="text/xsl" href="simple.xsl" ?>
<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"/>
ad { display: block; color: blue;}
year, make, model { color: red; font-size: 16pt;}
```

Ans:

- create table theater\_schedule (
 title varchar(20),
 performer varchar(40),
 date date
 );
- ```
<?xml version="1.0"?>
<theater_schedule>
  <program>
    <title>Saturday Night Live</title>
    <performer>Jayne Leno</performer>
    <date>2003-2-20</date>
  </program>
</theater_schedule>
```
- ```
<!DOCTYPE theater_schedule [
  <!ELEMENT theater_schedule (program)>
  <!ELEMENT program (title, performer, date)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT performer (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
]>
```

```

(d) theater_schedule
{
  background-color: #ffffff;
  width: 100%;
}
program
{
  display: block;
  margin-bottom: 30pt;
  margin-left: 0;
}
title, performer, date
{
  color: #FF0000;
  font-size: 20pt;
}

```

```

(e) <?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
        <table>
          <tr>
            <th>Title</th>
            <th>Performer</th>
            <th>Date</th>
          </tr>
          <xsl:for-each select="theater_schedule/program" order-by="+ date">
            <tr>
              <td><xsl:value-of select="title" /></td>
              <td><xsl:value-of select="performer" /></td>
              <td><xsl:value-of select="date" /></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>

```