PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO AT MISSISSAUGA
APRIL EXAMINATIONS 2005

CSC 309H1
Mississauga Campus

Duration — 3 hours

Examination Aids: ***Two double sided*** $8\frac{1}{2} \times 11$ ***aid sheets. A non-programmable calculator.***

Student Number: |_|_|_|_|_|_|_|_|_|_|_|

Last Name: _____

First Name: _____

Lecture Section: L5101
(A. Rosenbloom)

---

*Do **not** turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above, and read the instructions below *carefully*.)

This final examination consists of 7 questions on 16 pages (including this one), printed on both sides of the paper. *When you receive the signal to start, please make sure that your copy of the examination is complete.* Answer each question directly on the examination paper, in the space provided.

Be aware that concise, well thought-out answers will be rewarded over long rambling ones. Also, unreadable answers will be given zero (0) so write legibly.

General Hint: We were careful to leave ample space on the examination paper to answer each question, so if you find yourself using much more room than what is available, you're probably missing something. Also, remember that hints are just hints: you are not required to follow them if you can think of a different solution.

Good Luck!

1: _____/ 8
2: _____/10
3: _____/ 5
4: _____/10
5: _____/10
6: _____/10
7: _____/15

TOTAL: _____/68

Question 1. [8 MARKS]

Short Answers:

Part (a) [6 MARKS]

Circle T or F below depending on whether the statement is True or False

1. **T/F** A java console application can send cookies to a web server.
2. **T/F** Server side CGI scripts can only be written in perl.
3. **T/F** Servlets can be accessed only by servlet aware browsers.
4. **T/F** XML documents are designed to describe document appearance.
5. **T/F** An XSL stylesheet can be used to transform a CSV (comma separated values) document into an XML document.
6. **T/F** XML is is the result of database queries.

Part (b) [2 MARKS]

Circle the best response

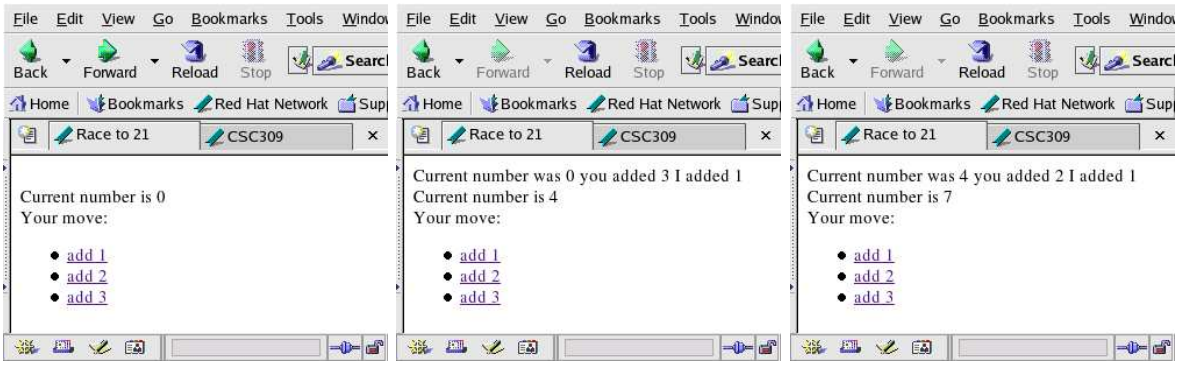
1. Servlets are able to maintain sessions because
   a) they use cookies to store the state of the application on the browser
   b) they use the servlet protocol
   c) they implement the session protocol
   d) they use hidden variables
   e) none of the above
2. The http protocol is implemented on top of
   a) TCP
   b) IP
   c) Ethernet
   d) SMTP
   e) none of the above

Question 2. [10 MARKS]

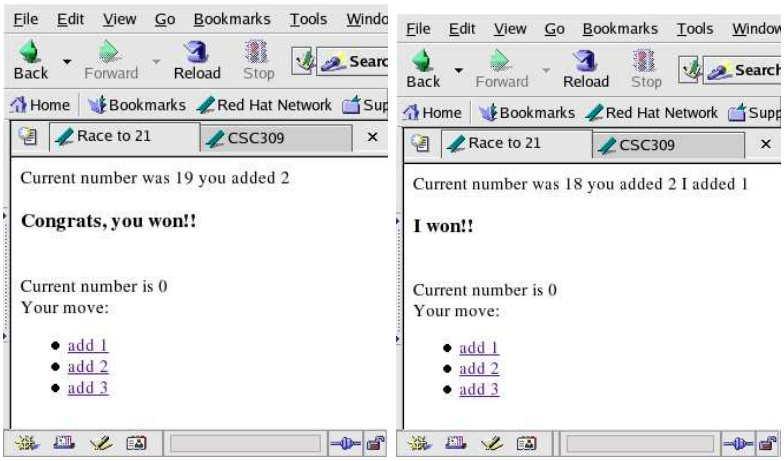
CGI/Perl: Create the perl CGI scripts described below.

Race to 21: The *Player* plays against the *Server* with the Player moving first. The game maintains a **currentNumber**. During their turn the Player/Server adds one of 1, 2 or 3 to the currentNumber. The participant that moves currentNumber to 21 wins the game.

Below is a sequence of screens in a game: The Player visits **http://www.gamesite.com/games/Race21Servlet.pl** (to get the left screen shot), next they click on the 'add 3' link, next they click on the 'add 2' link.



A Player and a Server winning screen are shown below.



Notes: You can use the CGI.pm perl module and should store the game state in the URLs. Each click on a link submits a request back to one of your CGI scripts. Best marks for a good computer strategy and simple script/scripts.

(continued...)

Question 3. [5 MARKS]

Do (parts 1 and 2 below) OR (part 3 below).

1. What is the purpose of CSS?
2. What is a JavaBean and how is the concept related to JSP?
3. In the last question, you created a perl script which plays RaceTo21. Create a Java program which uses sockets and wins the RaceTo21 game (causes the web server to return the *Congrats, you won!!* web page). **Hint:** Your java program does not have to actually play the game. You should be able to run your java program from any web connected computer.

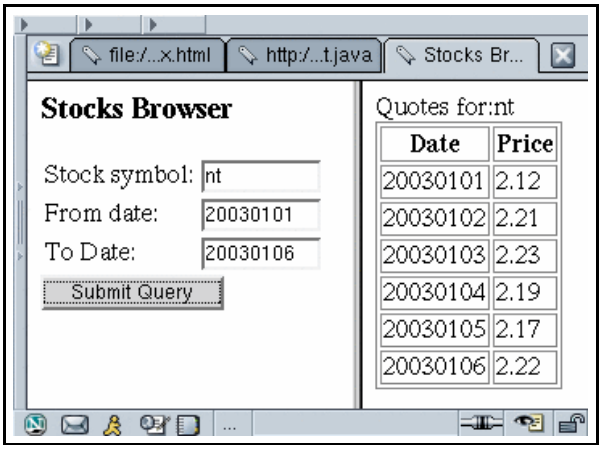
Question 4. [10 MARKS]

Create a Java Servlet for browsing historical stock quotes described below:

- Users specify a stock symbol, and an initial and final date.
- When the user presses the submit button, the Servlet returns a web page consisting of the name of the company and a table with the stocks daily prices for the specified period.
- Stock data is stored in a postgresql database with the following two tables.

```
CREATE TABLE stocks (
    id INTEGER PRIMARY KEY,
    symbol VARCHAR(3),
    company VARCHAR(20)
);
CREATE TABLE quotes (
    stockID INTEGER,
    quoteDate INTEGER,
    price REAL
);
```

- Your servlet generates the contents of the right frame shown below. Your servlet should not generate the form in the left frame.



- Do not worry about error handling.
- Assume that the user always provides all three values (stock symbol and dates) in a valid format
- Assume that the database contains records for the stock and dates the user is interested in.
- Dates are stored as 8 digit integers. For example, January 6, 2003 is stored as the integer 20030106.
- The postgresql database is named *stockExample* at *db.stockQuotes.com*, userId is *quoteClient*, password is *trespass*

(continued...)

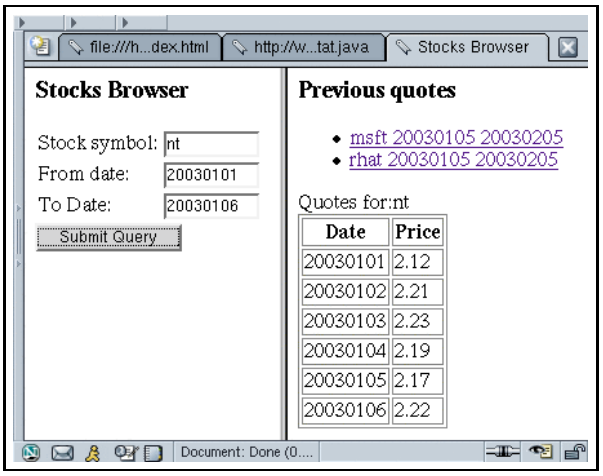
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import java.sql.*;

public class GetQuote extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
```

(continued...)

Question 5. [10 MARKS]

Modify your servlet above so that it tracks all previous quotes the user has requested **this session**. The user should be able to click on a past quote to retrieve it again.

You can mark locations on the previous page and place corresponding code here. You can add any new classes etc. here.



(continued...)

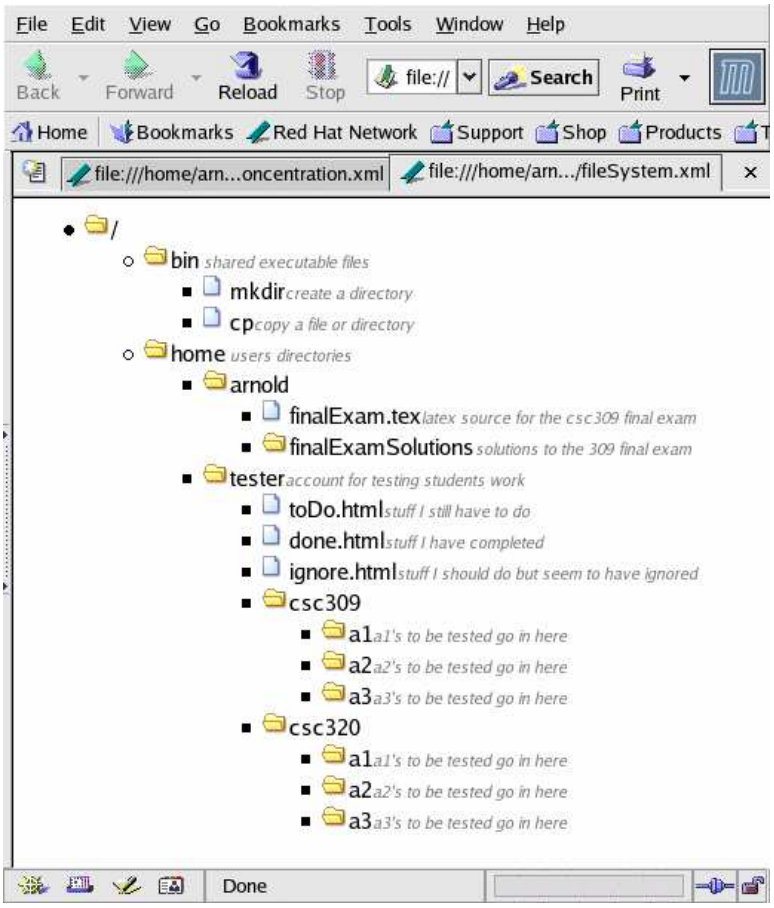
Question 6. [10 MARKS]

XML/XSL Create fileSystem.xsl as described below.

Below, you can find part of an instance of fileSystem.dtd. It captures information about files and directories in a file system. Permissions are written down using the notation common to unix systems. On the following page, you can find the result of applying fileSystem.xsl to the XML document below.

```
<!ELEMENT fileSystem (directory)>
<!ELEMENT directory (description?,permissions,(file|directory)*)>
<!ELEMENT file (description?,permissions) >
<!ELEMENT description (#PCDATA)>
<!ELEMENT permissions EMPTY>
<!ATTLIST permissions
        owner (rwx|-wx|r-x|rw-|r--|-w-|--x|---) "rw-"
        group (rwx|-wx|r-x|rw-|r--|-w-|--x|---) "---"
        world (rwx|-wx|r-x|rw-|r--|-w-|--x|---) "---"
>
<!ATTLIST directory name CDATA #IMPLIED >
<!ATTLIST file name CDATA #IMPLIED >
```

```
<fileSystem>
    <directory name="/">
        <permissions owner="rwx" group="r-x" world="r-x"/>
        <directory name="bin">
            <description> shared executable files </description>
            <permissions owner="rwx" group="r-x" world="r-x"/>
            <file name="mkdir">
                <description>create a directory</description>
                <permissions owner="rwx" group="r-x" world="r-x"/>
            </file>
            <file name="cp">
                <description>copy a file or directory</description>
                <permissions owner="rwx" group="r-x" world="r-x"/>
            </file>
        </directory>
        <directory name="home">
            <description> users directories </description>
            <permissions owner="rwx" group="r-x" world="r-x"/>
            <directory name="arnold">
                <permissions owner="rwx" group="---" world="---"/>
                <file name="finalExam.tex">
                    <description>latex source for the csc309 final exam</description>
                    <permissions owner="rw-" group="---" world="---"/>
                </file>
                <directory name="finalExamSolutions">
                    <description>solutions to the 309 final exam</description>
                    <permissions owner="rwx" group="---" world="---"/>
                </directory>
            </directory>
            <directory name="tester">
                <description>account for testing students work</description>
                <permissions owner="rwx" group="---" world="---"/>
                <file name="toDo.html">
                    <description>stuff I still have to do</description>
                    <permissions owner="rw-" group="---" world="---"/>
                </file>
                <file name="done.html">
                    <description>stuff I have completed</description>
                    <permissions owner="rw-" group="---" world="---"/>
                </file>
                <file name="ignore.html">
                    <description>stuff I should do but seem to have ignored</description>
                    <permissions owner="rw-" group="---" world="---"/>
                </file>
                ...
```



Your job is to write fileSystem.xsl. Assume that you have access to images *folderopen.gif* and *page.gif*.

(continued...)

Question 7. [15 MARKS]

Servlet Sessions: Java Servlets implements the notion of session. Describe how you could introduce the notion of session to perl CGI scripts by describing the implementation behind perl functions *getSession, updateSession, getFromSession* below.

- *getSession()* starts a new session for this client if one does not already exist.
- *updateSession(key,value)* associates the string value with the string key in the clients session.
- *getFromSession(key)* returns the value associated with key in the clients session.

Notes: As in the Java Servlet notion of Session, try not to store too much state information on the client. Your functions will be run by Perl/CGI scripts like the ones you wrote for assignment 2. Unlike Java sessions, you will just store strings in your sessions.

To get started, describe how sessions are to be implemented, then talk about each of the above functions.

There is NO question on this page!

*[If you need extra space to answer a question, use the space below and indicate **clearly** the question number.]*

There is NO question on this page!

*[If you need extra space to answer a question, use the space below and indicate **clearly** the question number.]*

Total Marks = 68