PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

APRIL EXAMINATIONS 2003

CSC 309H1
Mississauga Campus

Duration — 3 hours

Examination Aids: ***Three 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: _____/10
2: _____/10
3: _____/10
4: _____/10
5: _____/10
6: _____/ 5
7: _____/10

TOTAL: _____/65

Question 1. [10 MARKS]

Part (a) [4 MARKS]

LangerCorp.com has been assigned the class C address 198.77.116.0. and has decided to divide their address space into 4 equally sized subnets (s1,s2,s3,s4).

Complete the following table describing the four subnetworks:

```
Subnet s1: 198.77.116.0  -_________________    Subnetmask:___________________

Subnet s2: _____________-_________________    Subnetmask:___________________

Subnet s3: _____________-_________________    Subnetmask:___________________

Subnet s4: _____________-_________________    Subnetmask:___________________
```

Hints:

- s1 should consist of a contiguous block of IP addresses, similarly for s2, s3 and s4.
- **Network:** can be computed by **anding** the subnetmask and an IP address. For example, both of the following IP addresses are on the same class C network 210.5.7.0 (with subnetmask 255.255.255.0).

```
  210.  5.  7. 11          210.  5.  7.108
& 255.255.255.  0        & 255.255.255.  0
-----------------        -----------------
  210.  5.  7.  0    =     210.  5.  7.  0
```

Part (b) [6 MARKS]

Complete the following IP Routing table for host h1 (SEE NEXT PAGE)

```
Kernel IP routing table
Destination          Gateway              Genmask              Iface
```

```
210.5.7.0            0.0.0.0              255.255.255.0        e0
127.0.0.0            0.0.0.0              255.0.0.0            lo
```

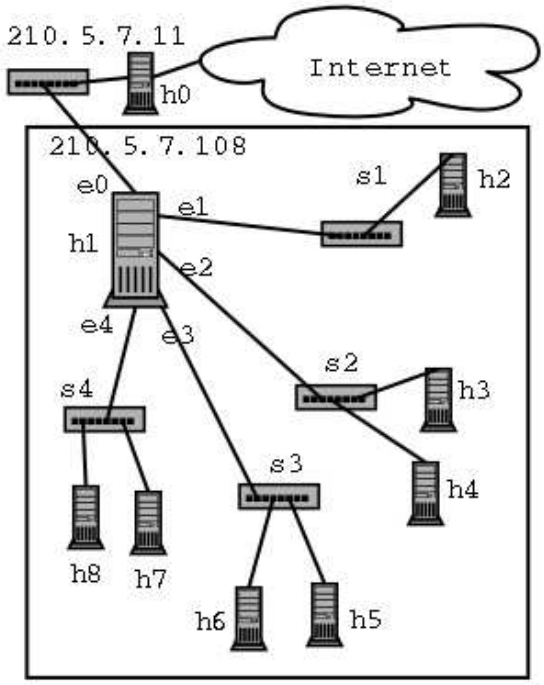


Figure 1: LangerCorps Network

- LangerCorp.com's main gateway (denoted **h1** in the diagram) is given IP address 210.5.7.108 on their ISP's 210.5.7.0 network. (ISP stands for Internet Service Provider)
- The ISP's main gateway is **h0** and has IP address 210.5.7.11 on the 210.5.7.0 network.
- LangerCorp.com has given IP 210.5.7.108 to interface **e0** on host **h1**.
- **Destination:** is the destination network
- **Genmask:** is the subnet mask
- **Iface:** is the interface (also known as ethernet card) to use
- The IP Routing table we discussed in class is shown below

```
Destination     Gateway         Genmask         Iface
192.168.1.0     0.0.0.0         255.255.255.0   eth1
142.150.8.0     0.0.0.0         255.255.252.0   eth0
127.0.0.0       0.0.0.0         255.0.0.0       lo
0.0.0.0         142.150.10.224  0.0.0.0         eth0
```

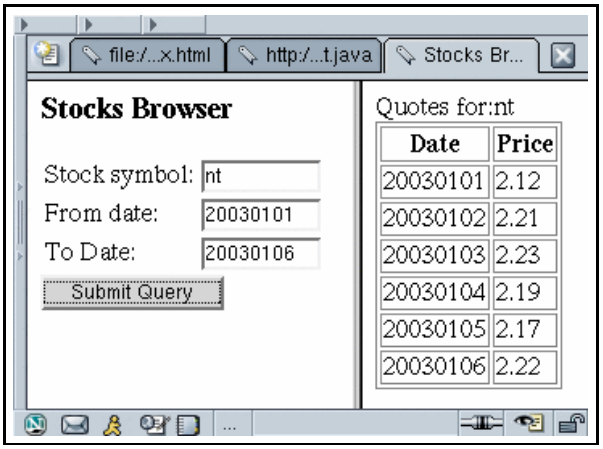
Question 2. [10 MARKS]

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...)

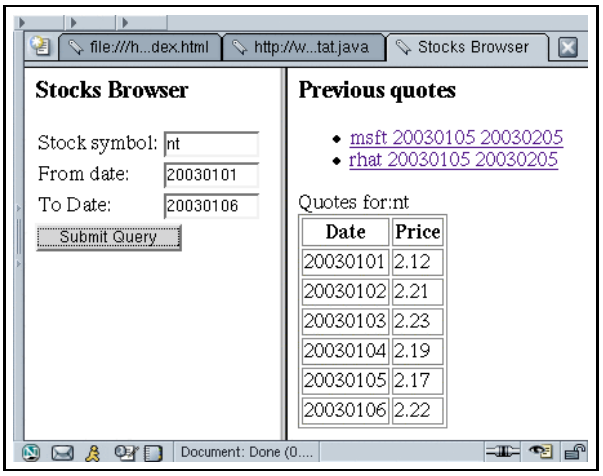
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import java.sql.*;

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 3. [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 4. [10 MARKS]

XML Consider the following dtd (*homes.dtd*)

```
<xmp>
<!ELEMENT Listing (Home*)>
<!ELEMENT Home (Address,Bedrooms,Price,LotSize?,Room+)>
<!ELEMENT Address (Street,City)>
<!ELEMENT Street (#PCDATA)>
<!ELEMENT City (#PCDATA)>
<!ELEMENT Bedrooms (#PCDATA)>
<!ELEMENT Bathrooms (#PCDATA)>
<!ELEMENT Price (#PCDATA)>
<!ELEMENT LotSize (#PCDATA)>
<!ELEMENT Room EMPTY>
<!ATTLIST Home id ID #REQUIRED>
<!ATTLIST Room
    type (kitchen|bedroom|bathroom|livingroom) #REQUIRED
    length CDATA #REQUIRED
    width CDATA #REQUIRED>
```

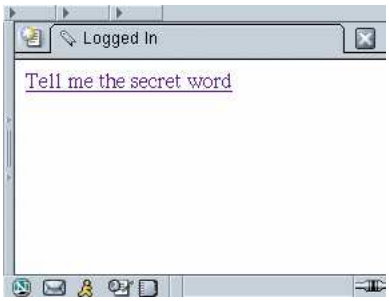
The XML document below is supposed to be an instance of the above DTD.
Mark all errors on the XML document below.

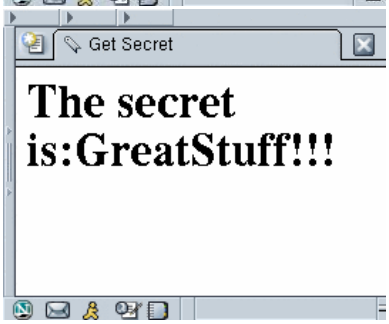
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE Listing SYSTEM "homes.dtd">
<Listing>
    <Home id="1">
        <Address>
            <Street> 20 Happy Street </Street>
            <City> Toronto </City>
        </Address>
        <Bedrooms> 3 </Bedrooms>
        <Bathrooms> 2 </Bathrooms>
        <Price> 250000 </Price>
        <Room type="kitchen" length=10 color=red>
            Updated appliances
    </Home>
    <home>
        <Address>
            <Street> 15 Sunny Avenue </Street>
            <City> Toronto </City>
            <Province>Ontario</Province>
        </Address>
        <Bedrooms> 2 </Bedrooms>
        <LotSize> 20*110 </LotSize>
        <Price> 150000 </Price>
    </Home>
</Listing>
```

Question 5. [10 MARKS]

CGI/Perl: Create the *secret.pl* CGI perl script described below.

- When *secret.pl* is given parameters *uid* and *password* (ie screen 1 below) then
  - if *(uid eq "arnold" && password eq "spiderman")* is true it returns a webpage with the link *Tell me the secret word* (ie screen 2 below) and at the same time, sets the cookie *loggedIn* to the value *"true"*
  - otherwise it simply reports that *Login failed!* (ie screen 3 below)
- If called with parameter *(getSecret eq "true")* (ie from the *Tell me the secret word* link) and the cookie *(loggedIn eq "true")* then the secret word (read from file secret.txt) is returned (ie screen 4 below) otherwise it returns *You are not logged in* (ie screen 5 below).



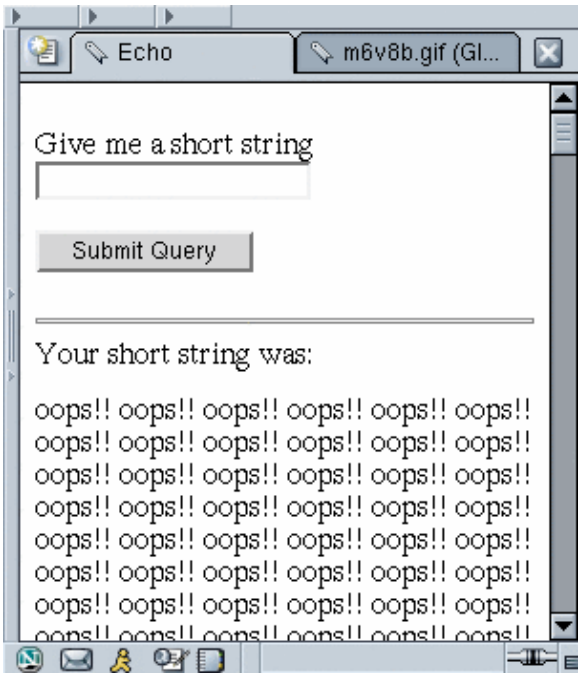


(continued...)

Question 6. [5 MARKS]

Consider *carefully* the CGI script written below. What string could be typed into the textfield to cause the web page below to appear?

```
#!/usr/bin/perl
use strict;
use CGI qw/:standard/;
print header, start_html('Echo'), start_form,
     "Give me a short string",textfield('ss'),p,submit, end_form, hr;
if (param()) {
        my $shortString=param('ss');
        if(length $shortString<100){
                print "Your short string was:<xmp>$shortString</xmp>";
        } else {
                print "Your string was not short!!";
        }
}
print end_html();
```



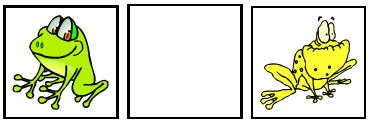
Question 7. [10 MARKS]

Consider *frogs.jsp* and *FrogsBean.java* on the following page with frogs.jsp available at the URL *http://localhost:8080/frogs/frogs.jsp*.

Describe the application at the above URL. Your high level description explains the application to non computer scientists (ie your kid brother) and includes answers to the following questions:

- When the user first visits the URL above, what do they see?
- What happens as a result of user interaction with the application? You should have statements of the form *When the user clicks on X, the result is Y.*
- When is the "Congrats" message displayed?

The following images are (from left to right) greenFrog.gif, empty.gif and yellowFrog.gif.



frogs.jsp

```
<jsp:useBean id="b" class="frogs.FrogsBean" scope="session"/>
<jsp:setProperty name="b" property="move"/>

<html>
<head><title>Frog Jumping Game</title></head>
<body bgcolor="white">
   <% String [] images={"greenFrog.gif","empty.gif","yellowFrog.gif"}; %>
   <table border><tr>
   <% for(int i=0;i<=6;i++){ %>
      <td><a href=frogs.jsp?move=<%= i %> >
         <img width=50 src=<%= images[b.get(i)+1] %>>
      </a></td>
   <% } %>
   </tr></table>
   <a href=frogs.jsp?move=-1>Start New Game</a>
   <% if(b.isSolved()){ %>
      Congrats, you solved it!!
   <% } %>
</body>
</html>
```

FrogsBean.java

```
package frogs;
public class FrogsBean {
   public static final int YELLOW=1, EMPTY=0, GREEN=-1;
   private int [] board=new int[7];

   public FrogsBean(){ reset(); }
   public void reset(){
      board[0]=board[1]=board[2]=YELLOW;
      board[3]=EMPTY;
      board[4]=board[5]=board[6]=GREEN;
   }
   public void setMove(String move){
      try {
         int m = Integer.parseInt(move);
         if(m==-1)reset();
         else if(!move(m,1))move(m,2);
      } catch (NumberFormatException e) { }
   }
   private boolean move(int m, int d){
      int dest=m+board[m]*d;
      if(0<=dest && dest <=6 && board[dest]==EMPTY){
         board[dest]=board[m]; board[m]=EMPTY;
         return true;
      }
      return false;
   }
   public boolean isSolved(){
      for(int i=0;i<6;i++){
         if(board[i]>board[i+1])return false;
      }
      return true;
   }
   public int get(int i){ return board[i]; }
}
```

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 = 65