UNIVERSITY OF TORONTO MISSISSAUGA
APRIL 2010 FINAL EXAMINATION
CSC 309H5S
Programming on the Web
Arnold Rosenbloom
Duration — 3 hours
Aids: **Two double sided $8\frac{1}{2} \times 11$ aid sheets.**

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

Last Name: ______________________

First Name: ______________________

Signature: ______________________

---

*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 3 questions on 18 pages (including this one), printed on both sides of the paper. You should also find 3 additional single sided pages of application screenshots attached. *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.

The University of Toronto Mississauga and you, as a student, share a commitment to academic integrity. You are reminded that you may be charged with an academic offence for possessing any unauthorized aids during the writing of an exam, including but not limited to any electronic devices with storage, such as cell phones, pagers, personal digital assistants (PDAs), iPods, and MP3 players. Unauthorized calculators and notes are also not permitted. Do not have any of these items in your possession in the area of your desk. Please turn the electronics off and put all unauthorized aids with your belongings at the front of the room before the examination begins. If any of these items are kept with you during the writing of your exam, you may be charged with an academic offence. A typical penalty may cause you to fail the course.

Please note, you CANNOT petition to RE-WRITE an examination once you have begun writing.

# 1:	______/13
# 2:	______/20
# 3:	______/40
TOTAL:	______/73

Good Luck!

Question 1. [13 MARKS]

http/cgi/cookies: Consider the python cgi scripts `cookie_count.py`, `cookie_login.py`, `cookie_logout.py` and `arnolds_cgi_lib.py` at the end of this exam. These scripts live at `http://utmcs.net/cgi-bin/`. These work together as follows: A user first logs into the application using a form that submits to `cookie_login.py`. After a successful login, each visit to `cookie_count.py` will return a webpage with the number of times they have previously visited `cookie_count.py`.

Part (a) [3 MARKS]

Write a URL which could be entered in a browser and cause the browser to successfully login to the application.

Part (b) [10 MARKS]

Write the text of a telnet session/telnet sessions which would cause `cookie_count.py` to return `You have been here 3 times before`.

Question 2. [20 MARKS]

Ajax/php/javascript: `aGuessGame.html`, `aGuessGame.js` and `aGuessGame.php` together form the pieces of the GuessGame Ajax application pictured in the attached screenshots. Your job is to complete the code below.

aGuessGame.html

```
<html>
        <head>
                <script language="javascript" src="aGuessGame.js"></script>
        </head>
        <body>
                <h3>I am thinking of a number between 1 and 10...</h3>
                <form name="guessForm">
                        Your guess:
                        <input name="userGuess" type="text"/>
                        <input type="button" value="go!" onclick="guess();">
                </form>
                <div id="guessResults"></div>
        </body>
</html>
```

Part (a) [10 MARKS]

Complete `aGuessGame.js`. All game state information should be kept on the server.

```
function GetXmlHttpObject() {
        var xmlHttp=null;
        try { xmlHttp=new XMLHttpRequest(); }
        catch (e) {
                try { xmlHttp=new ActiveXObject("Msxml2.XMLHTTP"); }
                catch (e)  { xmlHttp=new ActiveXObject("Microsoft.XMLHTTP"); }
        }
        return xmlHttp;
}
/* Use the following to parse the servers response
 * response=xmlHttp.responseText.split(":");
 * isHighLowCorrect=response[0]; // one of "high", "low", "correct"
 * numberOfAttempts=response[1];
 * numberGuessed=response[2];
 */
```

(continued ...)

Part (b) [10 MARKS]

Write `aGuessGame.php`.

```
<?php
```

```
    # Below is the servers responce to the client.
    # The final : is so we dont have to worry about trailing white space
    echo("$isHighLowCorrect:$numberOfAttempts:$guess:");
?>
```

Question 3. [40 MARKS]

Struts2: See the appendix for details regarding this question as well as the attached ColorMixer screenshots. In particular, the decorator `main.jsp` as well as `Colour.java` and `ColourMixer.java`.

Part (a) [10 MARKS]

CSS: `main.css + main.jsp` applied to `add.jsp` and `mix.jsp` produced the screenshots in the Appendix. Use the property:value pairs below to complete `main.css`, the style sheet used to style the **Struts2** `Color Mixer` application shown in the attached screenshots. You will need to come up with the selectors yourself. The decorator `main.jsp`, can be found in the **Struts2** Appendix at the end of this exam.

main.css

```
/*
background: #ACE;                  background:#ACE;
border: 2px solid #FFCC99;         border: solid #FFCC99;
color: #000;                       color: #000;
color: #fff;                       display: inline;
font: 13px Verdana, sans-serif;    font: bold 16px Fixed, monospace;
font: bold 25px Fixed, monospace;  height:20px;
left: 25px;                        left: 25px;
letter-spacing: 0.5em;             margin: -10px -10px 10px -10px;
padding: 10px;                     padding: 15px 0 5px 0px;
padding: 2px 10px; /* top and bottom, left and right */
position: absolute;                position: absolute;
right: 25px;                       text-align: center;
text-align: right;                 text-decoration: none;
text-transform: lowercase;         top: 25px;
top: 55px;                         white-space: nowrap;
*/
body {
    position: relative;
    margin: 0;
    padding: 0;
}
```

(continued...)

Part (b) [30 MARKS]

Struts2 core coding: I have supplied you with the Model components `Colour.java` and `ColourMixer.java`. Complete each section below.

- **[10 Marks]** Write `ColourAddAction.java`, the Struts2 action that is called as a result of submitting a new colour.

- **[3 Marks]** Write `add.jsp`. This is used (with `main.jsp`) to generate the **add colours** page. You can leave out the `DOCTYPE` element.

- **[5 Marks]** Write `mix.jsp`. This is used (with `main.jsp`) to generate the **mixed colours** page. You can leave out the `DOCTYPE` element.

- **[7 Marks] Sketch** any other details or Model/View and Controller components you feel are needed to make the application work. **For example:** it seems that a `ColourAddContinueAction` is needed. This is the action which fires as a result of clicking the **Add Colours** box or the **Continue adding colours** link. This makes the session instance of `ColorMixer` available, it's execute method does nothing. It does not validate its inputs.

- [**5 Marks**] Tie all of this together by writing the needed parts of `struts.xml`.

(continued...)

cgi scripts

arnolds_cgi_lib.py

```
#!/usr/bin/python
import os, re

def parse_attributes_and_values(s, split_symbol):
    dict={}
    if s is not None:
        attributes_and_values=s.split(split_symbol)
        for attribute_and_value in attributes_and_values:
            m=re.search("^\s*(.*)=(.*)$", attribute_and_value)
            if m:
                dict[m.group(1)]=m.group(2)
    return dict

# request[parameter_name]=parameter value
request = parse_attributes_and_values(os.environ.get("QUERY_STRING"),'&')

# cookie[cookie_name]=cookie value
cookie  = parse_attributes_and_values(os.environ.get("HTTP_COOKIE"),';')
```

cookie_login.py

```
#!/usr/bin/python
from arnolds_cgi_lib import request, cookie # this defines request and cookie

if "uid" in request: uid=request["uid"]
else: uid=""
if "password" in request: password=request["password"]
else: password=""

if uid=="arnold" and password=="spiderman":
    print "Content-Type: text/html"
    print "Set-Cookie: loggedIn=true; path=/"
    print ""
    print "<html>"
    print "<body>"
    print "<h3>Login success!</h3>"
    print "</body>"
    print "</html>"
else:
    print "Content-Type: text/html"
    print ""
    print "<html>"
    print "<body>"
    print "<h3>Login Failed!</h3>"
    print "</body>"
    print "</html>"
```

cookie_logout.py

```
#!/usr/bin/python
print "Content-Type: text/html"
print "Set-Cookie: loggedIn=false; path=/"
print ""
print "<html>"
print "<body>"
print "<h3>You are logged out!</h3>"
print "</body>"
print "</html>"
```

cookie_count.py

```
#!/usr/bin/python
from arnolds_cgi_lib import request, cookie # this defines request and cookie

if 'loggedIn' in cookie and cookie['loggedIn']=='true':
    if 'runningSum' in cookie:
        runningSum=cookie['runningSum']
    else:
        runningSum=0
    newRunningSum=1+int(runningSum)
    print "Content-Type: text/html"
    print "Set-Cookie: runningSum=" + str(newRunningSum)+"; path=/"
    print ""
    print "<html>"
    print "<body>"
    print "<h3>Counting with cookies</h3>"
    print "You have been here", runningSum, "times before."
    print "</body>"
    print "</html>"
else:
    print "Content-Type: text/html"
    print ""
    print "<html>"
    print "<body>"
    print "<h3>Counting with cookies</h3>"
    print "You are not logged in!"
    print "</body>"
    print "</html>"
```

Struts2 Appendix

Colour.java

```
package net.utmcs.colours;
public class Colour {
    private int red,green,blue;
    Colour(){ this(0,0,0); }
    Colour(int red, int green, int blue){
        this.setRed(red);
        this.setGreen(green);
        this.setBlue(blue);
    }
    public int getRed() { return red; }
    public void setRed(int red) {
        this.red = rangeTruncate(red);
    }
    public int getGreen() { return green; }
    public void setGreen(int green) {
        this.green = rangeTruncate(green);
    }
    public int getBlue() { return blue; }
    public void setBlue(int blue) {
        this.blue = rangeTruncate(blue);
    }
    private int rangeTruncate(int i){
        if(i<0)i=0;
        else if(i>255)i=255;
        return i;
    }
    public String getHexRepresentation(){
        String hex=Integer.toHexString(256*256*256+red*256*256+green*256+blue);
        return hex.substring(1);
    }
}
```

Struts2 Appendix

ColourMixer.java

```
package net.utmcs.colours;

import java.util.ArrayList;

public class ColourMixer {
    private ArrayList<Colour>colours=new ArrayList<Colour>();
    public void add(Colour c){ this.colours.add(c); }
    public void add(int r, int g, int b){ this.colours.add(new Colour(r,g,b)); }
    public ArrayList<Colour> getColours(){ return this.colours; }
    public Colour getAverage(){
        int r=0, g=0, b=0;
        for(Colour c:this.colours){
            r=r+c.getRed(); g=g+c.getGreen(); b=b+c.getBlue();
        }
        int size=this.colours.size();
        if(size>0)return new Colour(r/size,g/size,b/size);
        else return new Colour();
    }
}
```

main.jsp (the decorator template)

```
<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.1 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@taglib prefix="decorator" uri="http://www.opensymphony.com/sitemesh/decorator" %>
<%@taglib prefix="page" uri="http://www.opensymphony.com/sitemesh/page" %>
<%@taglib prefix="s" uri="/struts-tags" %>

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <title><decorator:title default="Colour Mixer"/></title>
        <link href="<s:url value='/styles/main.css'/>" rel="stylesheet"
        type="text/css" media="all"/>
        <decorator:head/>
    </head>

    <body>
        <div id="links">
            <a href="addContinue.action">Add Colours</a>
            <a href="mix.action">Mix Colours</a>
        </div>
        <div id="content">
            <decorator:body/>
        </div>
    </body>
</html>
```

Total Marks = 73