COMP/EECE 4081

Exam 1

Fall 2013

Name: Solutions

Rules:

- No potty breaks.
- Turn off cell phones/devices.
- Closed book, closed note, closed neighbor.

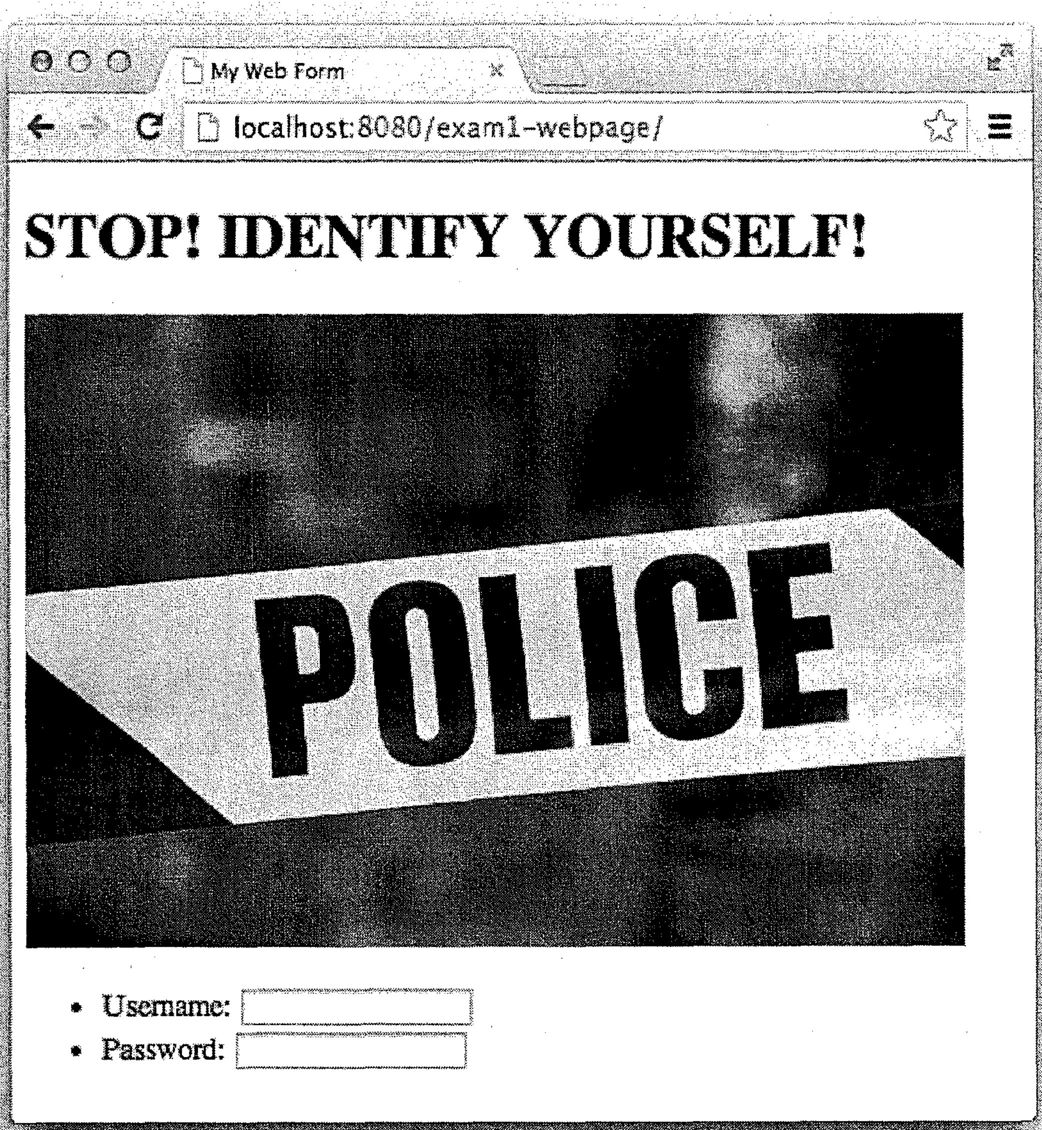
Reminders:

- Verify that you have 13 pages.
- Don't forget to write your name.
- Read each question carefully.
- Don't forget to answer every question.

Additional Items:

- For questions that involve writing code:
  - o You may omit `import` statements.
  - o You may omit exception-handling code.

1. [10pts] Write (on the next page) the complete HTML code that would create the web page depicted below. Your solution must include the following types of HTML elements (and no other types): **!DOCTYPE**, **body**, **h1**, **head**, **input** (with **type** and **name** attributes), **html**, **img** (with **src** attribute), **form** (with **action** and **method** attributes), **li**, **title**, and **ul**. The image is called **police-tape.jpg**. The form should send a POST request. The URL to which the form should be sent is **login.do**. Make the input field for "Username" a **text** field, and the one for "Password" a **password** field. The name for the username field is **loginUser**, and the name for the password field is **loginPwd**.



STOP! IDENTIFY YOURSELF!

- Username: [ ]
- Password: [ ]

**Note: You should ignore the fact that the form has no submit button (which is weird).

Write your answer to question 1 here.

```
<!DOCTYPE html>
<html>
<head>
<title>My Web Form</title>
</head>
<body>
<h1>STOP! IDENTIFY YOURSELF!</h1>
<img src="police-tape.jpg">
<Form action="login.do" method="post">
<ul>
 <li>Username: <input type="text" name="loginUser"></li>
 <li>Password: <input type="password" name="loginPwd"></li>
</ul>
</Form>
</body>
</html>
```

2. [2pts] What does a web browser do when you click a hyperlink? (Choose one.)

   a. Send an HTTP response

   b. Send an HTTP POST request

   [✓ c.] Send an HTTP GET request

   d. Send an HTTP WEB request

   e. None of the above

3. [2pts] What is a difference between HTTP GET and HTTP POST? (Choose one.)

   a. GET has a MIME type and POST does not

   b. GET has headers and POST does not

   [✓ c.] POST has a data payload and GET does not

   d. POST has a URL and GET does not

   e. None of the above

4. [2pts] Which of these Subversion client commands creates a working copy of the repository in the client's workspace. (Choose one.)

   a. update

   [✓ b.] checkout

   c. commit

   d. import

   e. None of the above

5. [2pts] True or False? In Subversion, if you want to make your working copy reflect any changes that have been made in the repository, you should run the **update** command.

   True

6. [2pts] In Subversion, which error might you get if you try to check-in a file that someone else has changed in the repository while you were working on your copy? (Choose one.)

   [✓ a.] Out of date

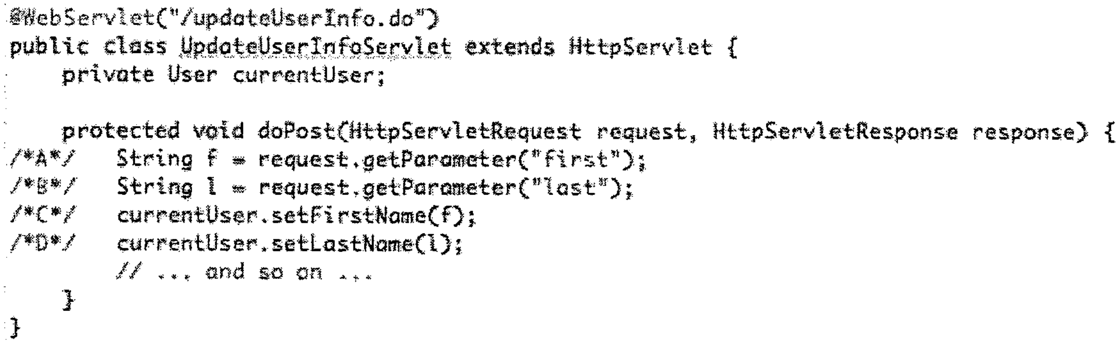
   b. 404

   c. 500

   d. Access denied

   e. 400 Bad request

7. [8pts] Imagine that two threads, T1 and T2, call the **doPost** method below concurrently. Fill in the table below to describe an execution scenario such that both threads execute all of lines A–D, and at the end, the **currentUser** has a first name given by T1 and a last name given by T2.

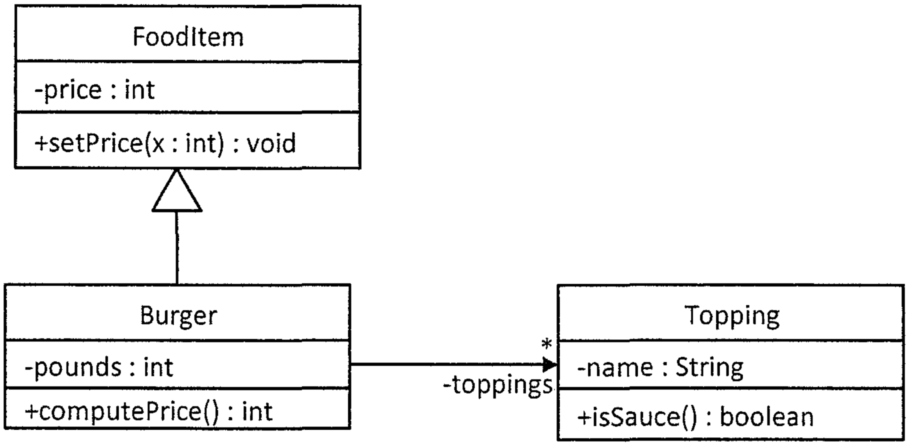


```
@WebServlet("/updateUserInfo.do")
public class UpdateUserInfoServlet extends HttpServlet {
    private User currentUser;

    protected void doPost(HttpServletRequest request, HttpServletResponse response) {
/*A*/   String f = request.getParameter("first");
/*B*/   String l = request.getParameter("last");
/*C*/   currentUser.setFirstName(f);
/*D*/   currentUser.setLastName(l);
        // ... and so on ...
    }
}
```

Executing thread (T1 or T2)	Line executed (A, B, C, or D)
T2	A
T2	B
T2	C
T1	A after
T1	B
T1	C
T1	D after
T2	D

8. [10pts] Create a Java code skeleton that is consistent with the following design class diagram. Do not implement the method bodies; instead, put "...". Use this page and the next one for your answer. Hint: You may want to use the ArrayList type in your answer (see Java Generics in the cheatsheet). Or maybe just use a plain old array. Either way is fine.



```
class FoodItem {
    private int price;
    public void setPrice(int x)  {...}
}
class Burger extends FoodItem {
    Private int pounds;
    public int computePrice()  {...}
    private ArrayList<Topping> toppings;
 }

class Topping {
    private String name;
    public boolean isSauce()  {...}
 }
```

Write the rest of your answer to question 8 here.

9. [18pts] Assume that there is a chess-game Java EE webapp with the following four classes/JSPs: **ChessBoard**, **MakeMoveServlet**, **board.jsp**, and **winner.jsp**. The program allows two players to make chess moves until one wins.

Your job is to write the code to implement **MakeMoveServlet** and **winner.jsp**. Implement only these two items – no additional classes or JSPs. Do not implement **board.jsp** and **ChessBoard** – assume they already exist. Also, implement only what is asked of you below. For example, assume the player's names have been set somehow, and that it is not your concern how.

Here are your constraints:

- Your implementation must follow the MVC architectural pattern.
- Use only the public interface for **ChessBoard**, given in Figure 1.
- Assume that **MakeMoveServlet** receives only POST requests, and those requests have three parameters:
  - "player" – A string with the player's ID.
  - "moveFrom" – The string describing the from location.
  - "moveTo" – The string describing the to location.
- If a player's move does not result in a win, then **board.jsp** should be displayed.
  - This JSP does not require any attributes from the servlet.
- If one player wins, then **winner.jsp** should be displayed.
  - This JSP must use a "winner" attribute that is a String with the winner's ID. (Hint: The word *attribute* in this sentence is significant.)
  - The displayed page should look like Figure 3.
- To help, the relevant parts of the servlet API have been given in Figure 2

Write your answer on the following 3 pages.

10. [6pts] For each of items from the previous question, tell what part of MVC they belong to. Spell the full name of each part (e.g., not just "M").

a. **ChessBoard** Model

b. **MakeMoveServlet** Controller

c. **winner.jsp** View

**Note: No URL pattern was specified for question 9, so you can make it whatever you want.

**Note: For the servlet class definition, you must include the @WebServlet tag and basic class header, but you may omit the inheritance and exception parts.

Write your answer to question 9 here.

```
@WebServlet("makeMove.do")
public class MakeMove Servlet extends HttpServlet {
      public void doPost(HttpServletRequest req,
                                 HttpServletResponce resp)
      {   String winner;
          String player = req[redacted].getParameter("player");
          String from = req[redacted].getParameter("moveFrom");
          String to = req[redacted].getParameter("moveTo");
          ChessBoard myCB = ChessBoard.getInstance();
          RequestDispatcher view;
          synchronized (myCB) {
                myCB.setTurn(player);
                myCB.move(from, to);
                winner = myCB.isCheckmate();
            }
            if (winner == null) {
                view = req[redacted].getRequestDispatcher("board.jsp");
                view.forward(req, resp);
            } else {
                req.setAttribute("winner", winner);
                view = req.getRequestDispatcher("wimer.jsp");
                view.forward(req, resp);
            }
      }
}
```

Write your answer to question 9 here.

```
<!DOCTYPE html>
<html>
<head>
<title>Winner!</title>
 </head>
 <body>
  <h1> Congrats,
        [redacted]
        <%= (String)request.getAttribute("winner") %> !
   </h1>
   </body>
    </html>
```

Write your answer to question 9 here.

Consider these relational database tables:

TVNetwork

NetworkID	NetworkName
111	CBS
666	Comedy Central
999	Fox

TVShow

ShowID	ShowName	Seasons
2222	The Simpsons	24
5555	Futurama	7
8888	Firefly	1

TVShowNetwork

ShowID	NetworkID
2222	999
5555	666
8888	999

Talent

TalentID	LastName	FirstName
333333	Groening	Matt
777777	Whedon	Joss

TVShowCreator

ShowID	TalentID
2222	333333
5555	333333
8888	777777

What would the following SQL queries return? Fill in the table below each query with your answer. Cross out any cells in the table that you do not need. Don't forget to label the columns.

**Correction: Should be "Seasons".

11. [3pts] `SELECT * FROM TVShow WHERE Season > 1;`

Show ID	ShowName	Seasons
2222	The Simpsons	24
5555	Futurama	7

12. [5pts] `SELECT LastName, ShowID FROM Talent INNER JOIN TVShowCreator`
`ON Talent.TalentID = TVShowCreator.TalentID`

Last Name	ShowID	
Groening	2222	
Groening	5555	
Whedon	8888	