

Student ID _____ Student Name _____

Web Applications Architecture and Frameworks DE

Midterm Exam May 18, 2013

PRIVATE AND CONFIDENTIAL

1. Allotted exam duration is 2 hours.
2. Closed book/notes.
3. No personal items including electronic devices (cell phones, computers, calculators, PDAs).
4. Cell phones must be turned in to your proctor before beginning exam.
5. No additional papers are allowed. Sufficient blank paper is included in the exam packet.
6. Exams are copyrighted and may not be copied or transferred.
7. Restroom and other personal breaks are not permitted.
8. Total exam including questions and scratch paper must be returned to the proctor.

3 blank pages are provided for writing the solutions and/or scratch paper. All 3 pages must be handed in with the exam

BE VERY CAREFUL WITH THE GIVEN 2 HOURS AND USE YOUR TIME WISELY. THE ALLOTTED TIME IS GIVEN FOR EVERY QUESTION.

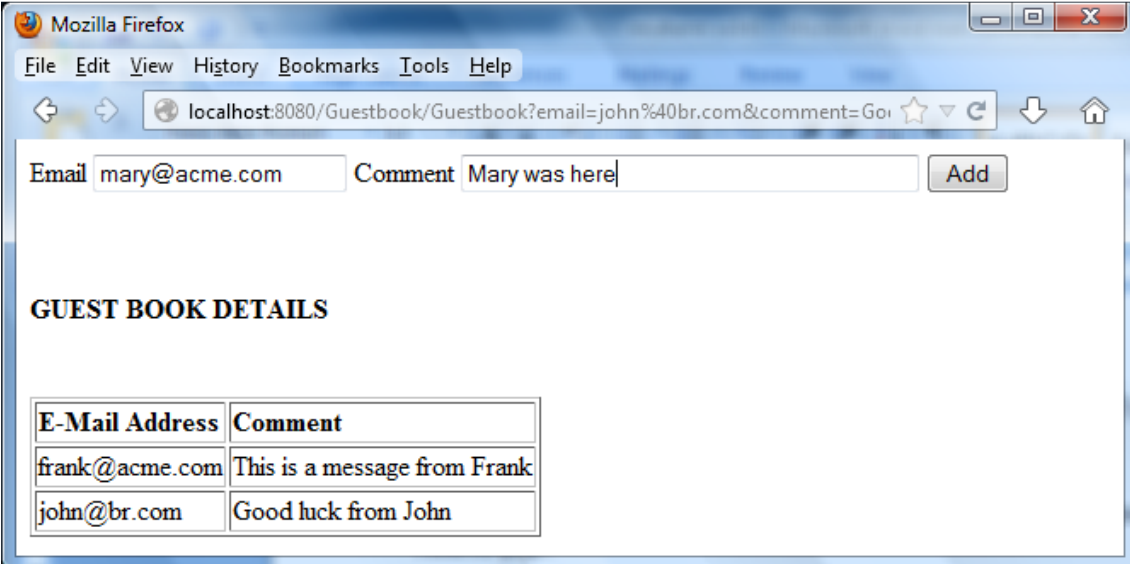
Write your name and student id at the top of this page.

Question 1: [10 points] {10 minutes}

- a. Explain all the differences between
 - `response.sendRedirect()`
 - `requestDispatcher.forward()`

Question 2. [30 points] {40 minutes}

Write the following guestbook application using one Servlet (Do not use HTML, JSP's or JSF!)



Email Comment

GUEST BOOK DETAILS

E-Mail Address	Comment
frank@acme.com	This is a message from Frank
john@br.com	Good luck from John

When you enter an email and comment and click the Add button, then this email and comment will be added to the table showing all guestbook entries. As a user of this application you should only see guestbook entries that you added yourself. You do not see guestbook entries added by another user.

Complete the partial given code. Do NOT write getter and setter methods!

```

@WebServlet(name = "Guestbook", urlPatterns = {"/Guestbook"})
public class Guestbook extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        List<Entry> entrylist = null;

        String email = request.getParameter("email");
        String comment = request.getParameter("comment");

        out.println("<Html>");
        out.println("<Form action= 'Guestbook' method= 'get'>");
        out.println("Email <Input Type='text' Name='email' Size='20'>");
        out.println("Comment <Input Type='text' Name='comment' Size='40'>");
        out.println("<Input Type='Submit' Value='Add'>");
        out.println("</Form><BR>");
        out.println("<Table border=1 >");
        out.println("<H4><B>GUEST BOOK DETAILS</B></H4><BR>");
        out.println("<TR><TD><b>E-Mail  
Address</b></TD><TD><b>Comment</b></TD></TR>");

        if (email != null) {
            HttpSession session = request.getSession();
            entrylist = (List<Entry>) session.getAttribute("entrylist");
            if (entrylist == null) {
                entrylist = new ArrayList<Entry>();
                session.setAttribute("entrylist", entrylist);
            }
            Entry newentry = new Entry(email, comment);
            entrylist.add(newentry);
            for (Entry entry : entrylist) {
                out.println("<TR><TD>" + entry.getEmail() + "</TD><TD>" + entry.getComment() +
                    "</TD></TR>");
            }
        }
        out.println("</Table>");
        out.println("</Html>");
    }
}

public class Entry {
    private String email;
    private String comment;
}

```

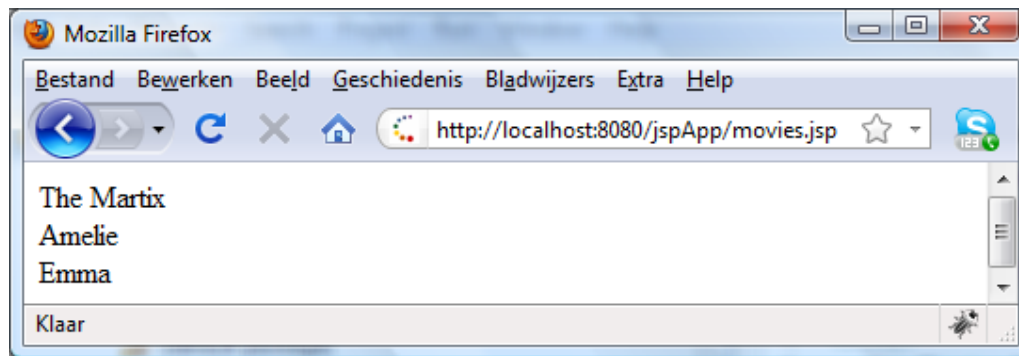
Question 3. [10 points] {10 minutes}

Complete the given JSP file below

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html>
<body>
  <% String[] movieList = {"The Martix", "Amelie", "Emma"};
  <% pageContext.setAttribute("movies", movieList);%>

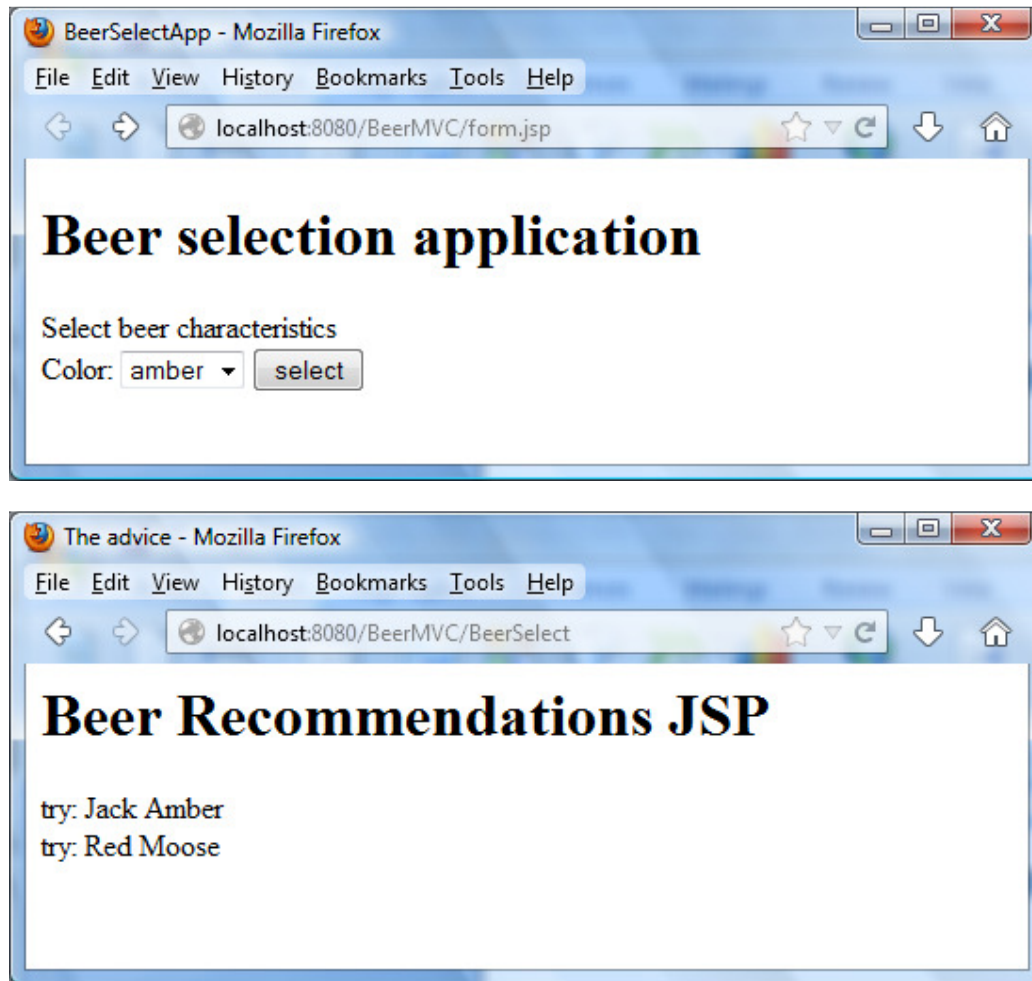
  <c:forEach var="movie" items="${movies}">
    ${movie}<br/>
  </c:forEach>
</body>
</html>
```

so that when we call this JSP file, we get the following webpage:



Question 4. [25 points] {30 minutes}

In chapter 3 of the book we used this course, the following MVC Beer Advisor application is discussed:



Complete the partial given code such that the code below works exactly as the Beer Advisor application pages given above. The application should follow the correct **Model-View-Controller** principles.

form.jsp:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <title>BeerSelectApp</title>
</head>
<body>
  <h1>Beer selection application</h1>
  <form method="POST" action="BeerSelect">
    Select beer characteristics<br/>
    Color:
    <select name="color" size="1">
      <option value="light">light</option>
      <option value="amber">amber</option>
      <option value="brown">brown</option>
      <option value="dark">dark</option>
    </select>
    <input type="submit" value="select">
  </form>
</body>
</html>
```

result.jsp:

```
<%@ page import="java.util.Iterator" %>
<%@ page import="java.util.List" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
  <title>The advice</title>
</head>
<body>
  <h1>Beer Recommendations JSP</h1>
  <%
    List brands = (List)request.getAttribute("brands");
    Iterator it = brands.iterator();
    while(it.hasNext()) {
      out.println("try: "+it.next()+"<br/>");
    }
  %>
</body>
</html>
```

BeerSelect.java

```
@WebServlet(name="BeerSelect", urlPatterns={"/BeerSelect"})
public class BeerSelect extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws
    IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String color = request.getParameter("color");
        BeerExpert beerExpert = new BeerExpert();
        if (color != null) {
            List brands = beerExpert.getBrands(color);
            request.setAttribute("brands",brands);
        }
        RequestDispatcher view = request.getRequestDispatcher("result.jsp");
        view.forward(request,response);
    }

}
```

BeerExpert.java

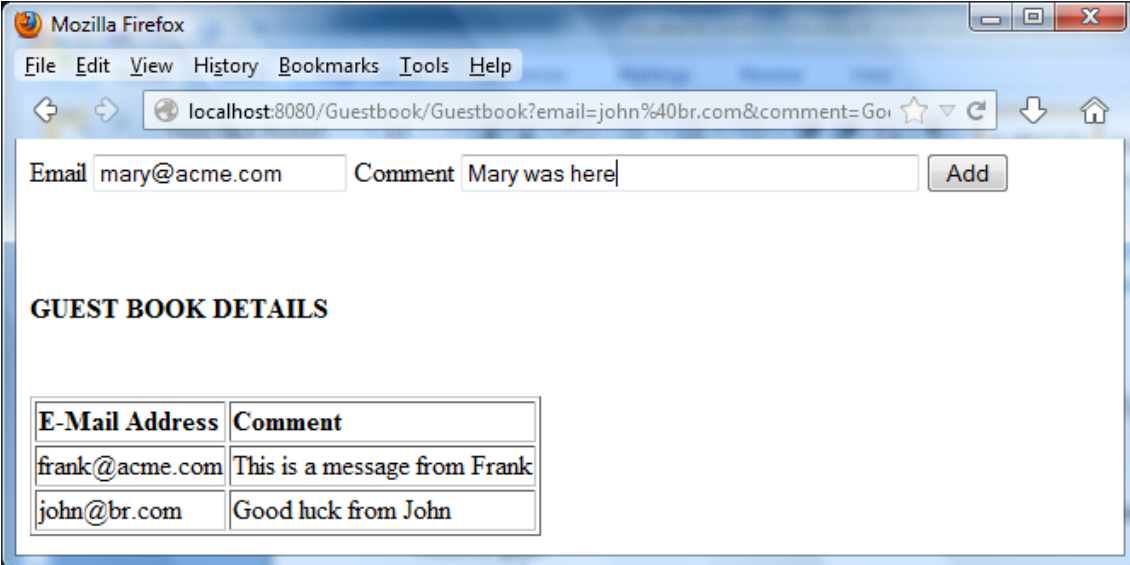
```
public class BeerExpert {

    public List getBrands(String color) {
        List brands = new ArrayList();
        if ("amber".equals(color)) {
            brands.add("Jack Amber");
            brands.add("Red Moose");
        } else {
            brands.add("Jail Pale Ale");
            brands.add("Gout Stout");
        }
        return brands;
    }

}
```

Question 5. [25 points] {30 minutes}

Write the following guestbook application using JSF (Do not use HTML, JSP's or servlets!)



Email Comment

GUEST BOOK DETAILS

E-Mail Address	Comment
frank@acme.com	This is a message from Frank
john@br.com	Good luck from John

When you enter an email and comment and click the Add button, then this email and comment will be added to the table showing all guestbook entries. As a user of this application you should only see guestbook entries that you added yourself. You do not see guestbook entries added by another user.

For this JSF application we will use annotation based configuration, so there is no faces-config.xml file. Add the necessary annotations to the code.

Complete the partial given code. Do NOT write getter and setter methods!

Guestbook.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html">
  <h:head>
    <title>Facelet Title</title>
  </h:head>
  <h:body>
    <h:form>
      Email <h:inputText value="#{guestbean.email}"/>
      Comment <h:inputText value="#{guestbean.comment}"/>
      <h:commandButton value="add" action="#{guestbean.add}"/>
    </h:form>
    <br/>
    <H4><B>GUEST BOOK DETAILS</B></H4><BR/>
    <h:dataTable border="1" var="entry" value="#{guestbean.entrylist}">
      <h:column>
        <h:outputText value="#{entry.email}"/>
      </h:column>
      <h:column>
        <h:outputText value="#{entry.comment}"/>
      </h:column>
    </h:dataTable>

  </h:body>
</html>
```

```
@ManagedBean
@SessionScoped
public class Guestbean {

    private List<Entry> entrylist = new ArrayList<Entry>();
    private String email;
    private String comment;

    public String add(){
        Entry entry = new Entry(email, comment);
        entrylist.add(entry);
        return null;
    }

}

public class Entry {
    private String email;
    private String comment;
}
```