

CS472 Web Programming Final Exam (A)

Professor Rujuan (Tina) Xing

Student Id: _____

Name: _____

The exam takes 2.5 hours. Total score is 100.

| 1-25 (50) | 26 (15) | 27 (25) | 28 (7) | 29 (3) | Total (100) |
|--------------|------------|------------|-----------|-----------|----------------|
| | | | | | |

Please read the exam policy before you start the exam.

Exam Policy:

There is no tolerance policy for exams. **You will be asked to leave the exam room immediately without a warning** once you do the following things which mean you'll get **NC**.

1. You are caught cheating or trying to cheat.
2. Answers should be written with a Pen or Pencil, but if you want to use a pencil please bring your own eraser and sharpener. You're not allowed to borrow from other students or proctors during exam.
3. All mobile phones should be turned off and submitted along with your luggage at the beginning of the exam.
4. Please get ready and drink water before the exam as **no one** will be allowed to leave the exam room before turning in their papers for water.
5. You're not allowed to ask/get extra papers from other students or proctors. All your notes must be written on the exam paper provided. Use the back side if you need to draft.

Please write down your answer clearly. If I cannot read your answer, you'll not get credit.

Good luck!

PART II(47 points): Programming Questions

1. **(15 points)** SimpleCalculator App: You're requested to develop a SimpleCalculator Application which asks for 3 parameters: 2 numbers and 1 operator. This application is only built by using Servlet **without JSP**.

When the application starts, it should automatically loads Servlet doGet() which displays the page below:

Simple Calculator

| | | | | | |
|----------------------|----------------------|----------------------|---|----------------------|--------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | = | <input type="text"/> | Submit |
|----------------------|----------------------|----------------------|---|----------------------|--------|

After input value into the text, click Submit button, it'll show the result on the last text. Assume the operator only allows "+", "-", "*", "/". You don't need to validate the operator, just assume user will input the correct value for operator. You must validate the input numbers are actually Integer.

Simple Calculator

| | | | | | |
|---|---|---|---|---|--------|
| 1 | + | 1 | = | 2 | Submit |
|---|---|---|---|---|--------|

If the input values are not numbers, just redirect back to current page.

web.xml – **TODO List**

- 1) Configure welcome page
- 2) Configure Servlet URL mapping. (You can use annotation too). One of them is fine.

```
<web-app>
  <welcome-file-list>
    <welcome-file>CalculatorServlet</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>CalculatorServlet</servlet-name>
    <servlet-class>com.wap.CalculatorServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>CalculatorServlet</servlet-name>
    <url-pattern>/CalculatorServlet</url-pattern>
  </servlet-mapping>
```

```
</web-app>
```

CalculatorServlet.java – TODO List

- 1) Configure Servlet URL by using annotation. (XML or annotation, one of them is enough)
- 2) Implement your logic in doGet() or doPost().

```
@WebServlet("/CalculatorServlet")
public class CalculatorServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        String num1 = request.getParameter("num1") == null ? "" :
request.getParameter("num1");
        String num2 = request.getParameter("num2") == null ? "" :
request.getParameter("num2");
        String operator = request.getParameter("operator") == null ? "" :
request.getParameter("operator");
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        StringBuffer sb = new StringBuffer();
        sb.append("<html><head><title>Insert title
here</title></head><body>");
        sb.append("<h1>Simple Calculator</h1>");
        sb.append("<form action='CalculatorServlet' method='post'>");
        sb.append("<input type='text' name='num1' value='" + num1 + "'/>
");
        sb.append("<input type='text' name='operator' size='1' value='" +
operator + "'/> ");
        sb.append("<input type='text' name='num2' value='" + num2 + "'/>
");
        sb.append(" = ");
        String result = "";
        if (Util.isNumeric(num1) && (Util.isNumeric(num2))) {
            int sum = 0;
            if ("+".equals(operator)) {
                sum = Integer.parseInt(num1) +
Integer.parseInt(num2);
            } else if ("-".equals(operator)) {
                sum = Integer.parseInt(num1) -
Integer.parseInt(num2);
            } else if ("*".equals(operator)) {
                sum = Integer.parseInt(num1) *
Integer.parseInt(num2);
            } else {
                sum = Integer.parseInt(num1) /
Integer.parseInt(num2);
            }

            result = String.valueOf(sum);
        }
        sb.append("<input type='text' name='sum' value='" + result + "'
/>");
        sb.append("<input type='submit' value='Submit' />");
        sb.append("</form></body></html>");
        out.println(sb.toString());
    }
}
```

```

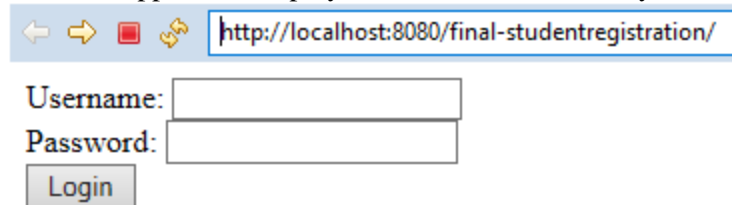
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}

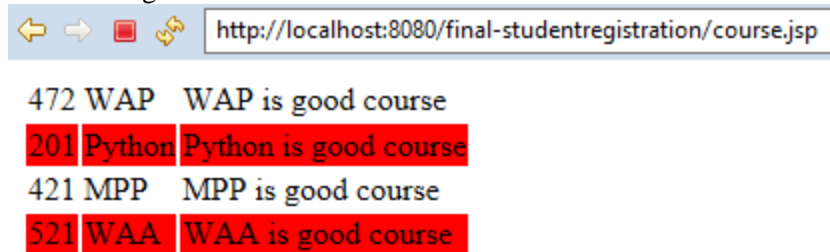
```

2. (25 points) Course Registration System: The Course Registration System is a simple application which can be used for students to check all registered course for authenticated users. It contains two pages: login.jsp – a simple form which used for user to login
course.jsp – a page which is used to display user's courses information
You also need to complete LoginServlet.java to fetch user's courses from CourseDatabase. Each course has id(Integer), name(String), Description(String) attributes.

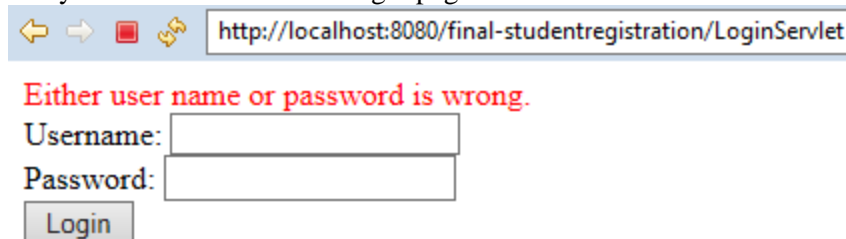
- 1) When the application deployed, it should automatically loads login.jsp page



- 2) After logged in, it redirects to course.jsp which displays courses. For **even rows**, it should have a background color red.



- 3) If the username and password isn't correct, it should shows login.jsp again with error message.
4) If user directly access <http://localhost:8080/final-studentregistration/course.jsp> without log in, the system should redirect to login page.



web.xml – **TODO List**

- 1) Configure welcome page
- 2) Configure Servlet URL mapping. (You can use annotation too). One of them is fine.

```

<web-app>
  <welcome-file-list>
    <welcome-file>login.jsp</welcome-file>
  </welcome-file-list>

```

```

</web-app>
login.jsp – Nothing to do in this page
<body>
  <form action="LoginServlet" method="post">
    Username: <input type="text" name="username"> <br />
    Password: <input type="password" name="password"><br />
    <input type="submit" value="Login">
  </form>
</body>
</html>

```

course.jsp – TODO List

- 1) Add necessary value in the taglib directive
- 2) You must use JSTL to loop all courses - forEach
- 3) For adding style, use JSTL core – if
- 4) You have to check if the user is authenticated to access this page, if not, dispatch to login.jsp page

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<html><head></head>
<style>
  .warning {
    background-color: red;
  }
</style>
</head>
<body>

<%
  //allow access only if session exists
  if (session.getAttribute("user") == null) {
    response.sendRedirect("login.jsp");
  }
%>
<table>
<c:forEach var="course" items="${courses}" varStatus="counter">

  <tr <c:if test="${counter.count % 2 == 0}">class="warning"</c:if>>
    <td>${course.id}</td>
    <td>${course.name}</td>
    <td>${course.description}</td>
  </tr>
</c:forEach>
</table>
</body>
</html>

```

Course.jsp – TODO List

- 1) Create a Java Class with name Course
- 2) This Course class is a **JavaBean** class.

//YOUR ANSWER

```
package com.wap;

import java.io.Serializable;

public class Course implements Serializable {

    private static final long serialVersionUID = -6604498309907045334L;
    private int id;
    private String name;
    private String description;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

}
```

CourseDatabase.java – Nothing to do for this class. Use this class to get courses for each user.

```
public class CourseDatabase {

    public List<Course> getCoursesByName(String username) {
        List<Course> courses = new ArrayList<>();
        //Assume here will to to Database to fetch courses for you
        return courses;
    }

}
```

CourseServlet.java – TODO List

- 1) Configure Servlet URL by using annotation. (XML or annotation, one of them is enough)
- 2) Implement code in doGet() or doPost()
- 3) Use USERNAME and PASSWORD to authenticate user

```
@WebServlet("/LoginServlet")
public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    private final String USERNAME = "admin";
    private final String PASSWORD = "admin123";

    CourseDatabase db = new CourseDatabase();

    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse
     * response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {

        // get request parameters for userID and password
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        if (USERNAME.equals(username) && PASSWORD.equals(password)) {
            HttpSession session = request.getSession();
            session.setAttribute("user", username);
            // setting session to expiry in 30 mins
            session.setMaxInactiveInterval(30 * 60);
            System.out.println(db.getCoursesByName(username));
            session.setAttribute("courses",
db.getCoursesByName(username));
            Cookie userName = new Cookie("user", username);
            userName.setMaxAge(30 * 60);
```

```
        response.addCookie(userName);
        response.sendRedirect("course.jsp");
    } else {
        RequestDispatcher rd =
getServletContext().getRequestDispatcher("/login.jsp");
        PrintWriter out = response.getWriter();
        out.println("<font color=red>Either user name or password
is wrong.</font>");
        rd.include(request, response);
    }
}
```


3. (7 points) You developed a dynamic web project named “final-ajax” which deployed on glassfish in your local machine. You’re asked to build an AJAX call to show greeting messages after user type their username. The only thing you need to do for this app is attach handler which makes the AJAX call and display the message in div with `id="ajaxGetUserServletResponse"` in main.js file. index.jsp – Nothing to do in this file

```
<html>
<head>
<script src="https://code.jquery.com/jquery-1.12.4.min.js"
  type="text/javascript"></script>
<script src="js/main.js" type="text/javascript"></script>
</head>
<body>
  <form>
    Enter Your Name: <input type="text" id="userName" />
  </form>
  <br>
  <br>
  <strong>Ajax Response</strong>:
  <div id="ajaxGetUserServletResponse"></div>
</body>
</html>
```

UserServlet.java – Nothing to do in this file

```
public class UserServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request,
        HttpServletResponse response) throws ServletException, IOException {

        response.setContentType("text/plain");
        response.getWriter().write("Hello, Tina");
    }
}
```

main.js – **TODO List**

- 1) Attach onblur event handler to input `type="text"`
- 2) Make an AJAX call to UserServlet
 - a. You have to pass user’s input to server side
 - b. It’s an POST call
 - c. If success, insert returned message to div
 - d. If fail, pop up “failed call” message.

//YOUR ANSWER

```
$(function() {
    $('#userName').blur(function(event) {
        var name = $('#userName').val();
        $.get('UserServlet', {
            userName : name
        }).done(function(responseText) {
            $('#ajaxGetUserServletResponse').text(responseText);
        }).fail(function() {
            //failed call message
        });
    });
});
```

```
$('#userName').blur(function() {
    $.ajax("UserServlet", {
        "type" : "GET",
        data : {
            userName : $('#userName').val()
        }
    }).done(function(greeting) {
        $('#ajaxGetUserServletResponse').text(greeting);
    }).fail(ajaxFailure);
});

$.ajax({
    url : 'UserServlet',
    data : {
        userName : $('#userName').val()
    },
    success : function(greeting) {
        $('#ajaxGetUserServletResponse').text(greeting);
    },
    error : ajaxFailure
});
```

4. (3 points) Write one or two paragraphs relating a point from the course to a principle from SCI.