

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"/>	<input type="button" value="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

<input type="text" value="1"/>	<input type="text" value="+"/>	<input type="text" value="1"/>	=	<input type="text" value="2"/>	<input type="button" value="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

Username:

Password:

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

472	WAP	WAP is good course
201	Python	Python is good course
421	MPP	MPP is good course
521	WAA	WAA is good course

- 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.

Either user name or password is wrong.

Username:

Password:

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="#" %>
```

```
<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 go 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="ajax GetUserServletResponse"` 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="ajax GetUserServletResponse"></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) {
            $('#ajax GetUserServletResponse').text(responseText);
        }).fail.ajaxFailure();
    });
});
```

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

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

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

Name: Dinh Tan Luong

StudentID: 985408

96.85

W1 Exam

CS472 WAP October 2016 - Prof. Zijlstra

1. [3 pts] Describe what the hosts file does, and when it is used:

+3

- It works like a local DNS in a computer, it maps domain names to IP addresses.
- When you connect to website via domain name, computer will look for it in the hosts file first. It can be used to map a domain name to localhost web server.

2. [3 pts] What is the difference between the internet and the World Wide Web?

+3

- Internet is a network where a single device like computer, mobile phone can join or leave it. It uses IP to connect between devices.
- World Wide Web is a set of websites or resources that can be accessed via internet.

3. [3 pts] What do the following HTTP codes mean?

200	OK ✓✓
403	forbidden
500	internal server error ✓

+3

4. [3 pts] Write a relative URL that would take you from

<http://mumstudents.org/~mzijlstra/test/index.html> to:

<http://mumstudents.org/cs472/params.php>

+3

.. / .. / cs472 / params . php

5. [3 pts] Describe the difference between display: none; and visibility: hidden;

+3

- display: none will hide the element and the space that it occupies placed.

- and visibility: hidden will hide the element but keep the space that it's placed. ex:

+3

6. [3 pts] Describe what the placeholder attribute does on an <input> tag:

placeholder is like a suggesting text inside the input. It will disappear when user types the real text in the input.
It cannot replace the input's value.

Name: Dinh Tan Liumg StudentID: 985408

7. [3 pts] What is the purpose of the <label> tag?

+3 when we use the <label> tag associated with the other input, Users can activate the input by clicking on the label.

ex: <label text=" for="name"> <input type="text" id="name" /> fname

8. [3 pts] Write a regular expression (regex) to match Iowa License Plates, these can be either 3 characters a space and then 3 numbers, or 3 numbers and then 3 characters

+3

$([A-Z]\{3\} \backslash d\{3\}) | (\backslash d\{3\} [A-Z]\{3\})$
 $([A-Z]\{3\} \backslash d\{3\}) | (\backslash d\{3\} [A-Z]\{3\})$

9. [3 pts] Write a regex that matches any line that starts with the word Exam followed by a number

+3

$^{\text{Exam}} [0-9]\{1,\}$
or: $^{\text{Exam}} \backslash d+$

10. [3 pts] Describe what SQL injection is, and how it can be prevented

- When a programmer uses string concatenation to make a SQL command, a hacker can use SQL injection to attack the website.

+3

- In java, we can use the prepared statement to prevent SQL injection.

11. [15 pts] On the next piece of paper (page 3) write HTML that would look like the screenshot below. Be sure to look at the next exercise (page 4 CSS) before you begin writing the HTML.

+15

Example page!

Two paragraphs with some text

And a table with headers and data:

Food	Tasty
Pizza	Very!
Ice Cream	Super!

Name: Dinh Tan Luong StudentID: 985408

Write your HTML here; please write a full page that will validate as correct HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset = "utf-8"/>
    <title> Example </title>
  </head>
  <body>
    <div id = "container">
      <h1> Example Page! </h1>
      <p> Two <em> paragraphs </em> with some <strong> text </strong> </p>
      <p> And a table with headers and data </p>

      <table>
        <tr>
          <th> Food </th>
          <th> Tasty </th>
        </tr>
        <tr>
          <td> pizza </td>
          <td> Very! </td>
        </tr>
        <tr>
          <td> Ice Cream </td>
          <td> Super! </td>
        </tr>
      </table>
    </div>
  </body>
</html>
```

Name: Dinh Tan Luong StudentID: 985408

- +15
12. [15 pts] Write CSS below to make the HTML from the previous page to look like the screenshot shown below. It is about 400px wide, light grey background, centered in the page, plus additional details shown in the image (colors and borders).

The screenshot shows a dark header with the text "Example page!". Below it are two paragraphs of text: "Two paragraphs with some text" and "And a table with headers and data:". A table follows, consisting of three rows and two columns. The first row has a header-like appearance. The second row contains "Pizza" in the first column and "Very!" in the second. The third row contains "Ice Cream" in the first column and "Super!" in the second.

Food	Tasty
Pizza	Very!
Ice Cream	Super!

```
#container {  
    width: 400px;  
    background-color: lightgrey;  
    border: 1px solid black; ✓  
    border-radius: 10px;  
    margin: 0 auto;  
}  
  
h1 {  
    background-color: black; ✓  
    color: white;  
    text-align: center;  
}  
  
p {  
    text-align: center; ✓  
}  
  
table {  
    margin: 0 auto; ✓  
}  
  
table th, table td {  
    text-align: center; ✓  
    border: 1px solid black;  
}  
  
        th {  
            font-weight: bold; ✓  
        }
```

Name: Dinh Tan Luong StudentID: 985408

9.6

13. [12 pts] What is the color:

<pre><body> <div id="first">First <li class="this that"> Second Third <li class="such"> Fourth <em id="so">Fifth </div> </body></pre>	<pre>body { background-color: ivory; } #first { color: blue; } #such { color: yellow; } #so, .that { background-color: lightblue; } ul, .that { background-color: white; } li.this.that { color: red; } .that strong { background-color: pink; } .such > em { color: purple; } div { color: green; }</pre>
--	---

	Foreground	Background
First	blue ✓	ivory ✓
Second	red ✓	pink ✓
Third	red ✓	white ✓
Fourth	green ✗	white ✓
Fifth	purple ✓	white ✗

14. [10 pts] What is the distance from the left most border of #first to the right most border of

+10 #second 170px

```
<body>
  <div id="first"></div>
  <div id="second"></div>
</body>

#first {
  float: left;
  margin: 20px 10px;
  border: 2px solid blue;
  padding: 10px 5px;
  height: 50px;
  width: 50px;
  background-color: #DDDD00;
}

#second {
  float: left;
  margin-left: 30px;
  margin-right: 10px;
  border: 3px solid green;
  padding: 10px 4px 10px 6px;
  height: 50px;
  width: 50px;
  background-color: #DD00DD;
}
```

first:
b: 2
p: 10 5
w: 50
mr: 20 10
p: 10 5
b: 2

74

second:
b: 3
p: 6
w: 50
ml: 30
p: 4
b: 3

96 5

margin is not collapsed ✓

+

= 170

Name: Dinh Tan Luong StudentID: 985408

15. [12 pts] Draw 4 layouts based on the following HTML, assuming a 1920x1080 screen

```
<body>
  <div id="a">Text for a
    <div id="b">Text for b
      <div id="c">Text for c</div>
    </div>
    <div id="d">Text for d</div>
  </div>
</body>
```

- a. Draw in the top left box the HTML with only with the following CSS:

+ 3

```
div { border: 1px solid black; }
```

- b. Draw in the top right box the CSS of a plus:

2.25

```
#a { width: 1000px; height: 500px; }
#b { position: fixed; bottom: 0px; left: 0px; right: 0px; }
#c { float: right; }
```

- c. Draw in the bottom left box the CSS of a and b plus:

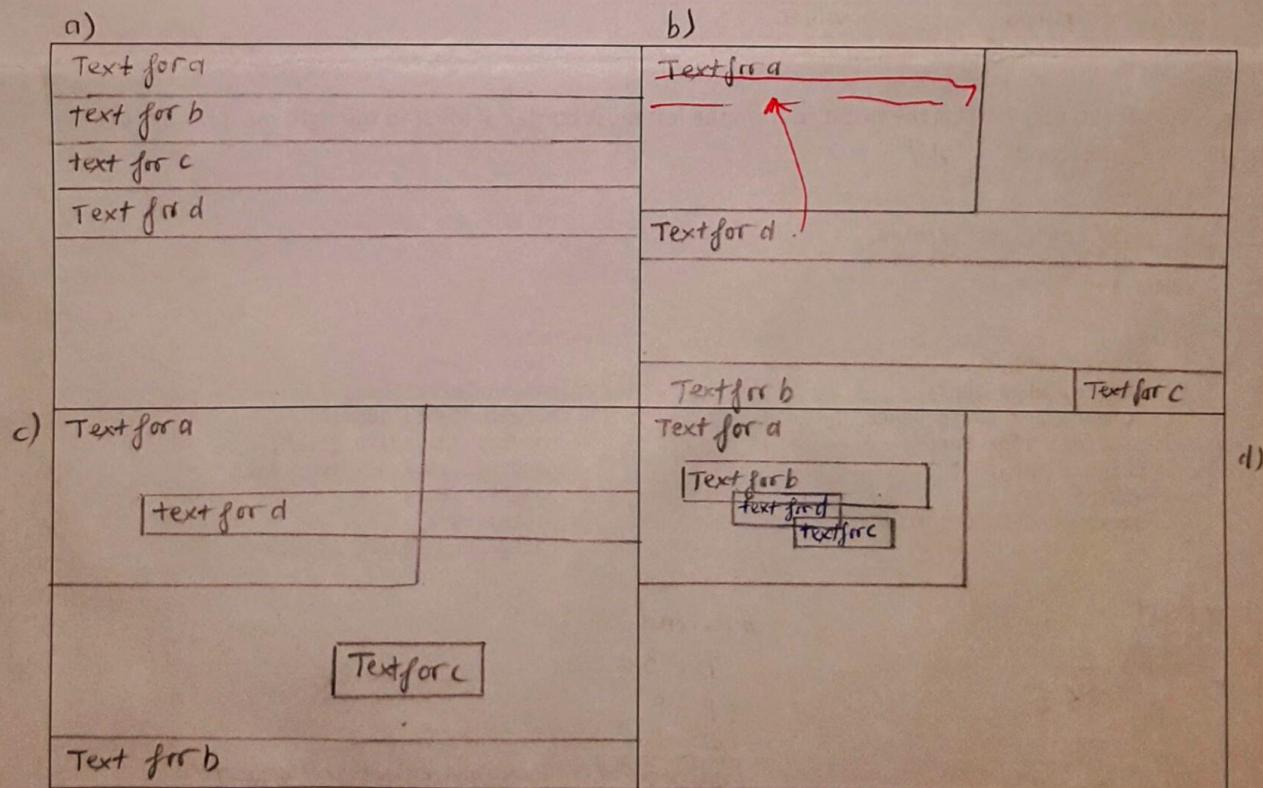
3

```
#a { position: relative; }
#c { float: none; position: absolute; left: 50%; bottom: 250px; }
#d { position: relative; top: 250px; left: 250px; }
```

- d. Draw in the bottom right box, the CSS of a, b, and c plus:

3

```
#b { position: static; margin: 100px; }
#d { position: absolute; }
```



Name: Dinh Tan Luong

StudentID: 985408

93.5

W2 Exam

CS472 WAP October 2016 - Prof. Zijlstra

1. [3 pts] Is the following code vulnerable to SQL injection? Please explain why:

```
public boolean addUser(Connection con, String name, String token) throws SQLException {  
    String sql = "INSERT INTO user VALUES(NULL, ?, '"+token+"')";  
    try (PreparedStatement ps = con.prepareStatement(sql)) {  
        ps.setString(1, name);  
        if (ps.executeUpdate() == 0) {  
            return false;  
        }  
    }  
    return true;  
}
```

+3

Yes. Because the sql statement uses string concatenation for the token.

2. [3 pts] Describe what a XSS attack is, and how you can defend against it:

+3

- It attacks by inserting a bad/evil script into the system.
- To defend against it, using escapeHTML before saving to the system.

3. [3 pts] What attack can be defended against by using Form Tokens:

+3

Cross Site Request forgery

4. [3 pts] Explain what Hoisting is in JavaScript:

+3

- When define & calling a variable or a function, JavaScript will look for its definition (declaration) in its local scope first, then global scope. If,
- for variable, if it was called before declaration, its value will be "undefined". If there is no declaration in both local and global, it will throw an error. We can use let or const to avoid hoisting.

5. [3 pts] Describe the difference between null and undefined

+3

- Null: the variable already know its type but has no value yet.
- Undefined: the variable has no value and doesn't know its type.

Name: _____ StudentID: _____

6. [3 pts] Does JavaScript support function overloading? Explain your answer:

- JavaScript does not support function overloading.
- Because the parameters for the function can be passed or not. If not, it will be undefined.

+3

ex: function log(var1, var2);
when we call log(var1); //var2 will be undefined.

7. [3 pts] Why is it considered to be a best practice to not include function definitions inside constructor functions, but instead put them on the Constructor.prototype property?

- Because the function may not need to be existed in every instance of created object.

+3

The function can be defined when a new object need it.

8. [3 pts] Explain what function currying is:

create new func with default param val
X not apply

- We can use .bind or .apply to change the way we call the function with its parameters.
- The parameters can be changed, added, reduced... like "currying".

+2.5

9. [3 pts] Explain what happens if you return false in an event handler attached with jQuery

- It will prevent browser from performing the usual action in response to the event like event.preventDefault

+3

And it also prevent the event from bubbling up further like event.stopPropagation().

10. [3 pts] Explain the difference between localStorage and sessionStorage

- session Storage:

+ Specifically for that tab.

+ when a tab is closed, the session storage (for that tab) is removed.
+ other opened tabs on the same domain cannot access its data.

+3

- local Storage:

+ Store more permanently on the client computer

+ Multiple tabs on the same domain can access it.

+ Not removed when tabs closed.

Name: Dinh Tan Luong StudentID: 985408

+ 8

11. [10 pts] What is the output of the following JavaScript:

```
var x = 5;
var y = 0;
function a(n) {
    var b = function() {
        console.log(this);
        return n;
    };
    b();
    console.log(y);
    if (n > 0) {
        x--;
        n *= 2;
    }
    var y = y - 1;
    console.log("x:" + x + " y: " + y);
    return b;
}
var f = a(x);
console.log(f.call({'z':100}));
```

Handwritten annotations:

- window ✓
- undefined ✓
- X: 4 y:-1 ✓
- // result for second statement ✓
- Object ✓
- undefined
- X: 3 y:-1 X

12. [5 pts] Which variables in the previous exercise can be considered free variables, which will be bound by an enclosure? Explain your answer:

25

- X is a free variable because x is used global and used in the n is a free variable for b function.
- x is also bound because x is assigned a reference to function f.
- y is also a free variable because for this line var y = y - 1; no that's just local variable

13. [5 pts] Write a JavaScript function called average() that takes a variable amount of numbers and returns the average of all the number given.

+ 5

```
function average()
{
    var avg = 0;
    var sum = 0;
    if (arguments.length > 0)
    {
        for (let i = 0; i < arguments.length; i++)
        {
            sum += arguments[i];
        }
        avg = sum / arguments.length;
    }
    return avg;
}
```

Name: _____ StudentID: _____

11.5

14. [12 pts] Rewrite the following obtrusive JavaScript to be unobtrusive without using jQuery, and then with using jQuery

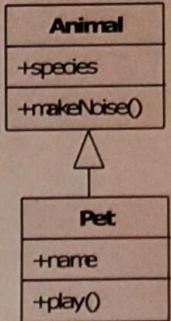
```
<a class="link" href="http://otherplace.com" onclick="return confirmLeave()">  
    See other place  
</a>  
<form id="form" onsubmit="return validateInput()">  
    ...  
    <input type="submit" />  
</form>  
<a class="link" href="http://otherplace.com" onclick="return confirmLeave()">  
    See other place  
</a>  
<input id="show" type="button" value="Show" onmouseover="showHidden()"/>  
<div id="hidden">Hidden stuff</div>  
  
<a class="link" href="http://otherplace.com" onclick="return confirmLeave()">  
    See other place  
</a>
```

Normal DOM, without jQuery	With jQuery
<pre>function confirmLeave() { ... } window.onload = function () { var links = document.getElementById .getElementsByTagName("a"); for(let i=0; i < links.length; i++) { links[i].onclick = confirmLeave; } document.getElementById("form"). onsubmit = validateInput; document.getElementById("show") .onmouseover = showHidden; }</pre>	<pre>\$(function () { \$(".link").click(confirmLeave); \$("#form").on("submit", validateInput); \$("#show").mouseover(showHidden); });</pre>

Name: Dinh Tan Luong StudentID: 985408

- +11
 15. [12 pts] Implement the following classes using JavaScript and create two pets, one dog called Fido, and one cat called Mimi. .makeNoise() should log: "The "+species + " makes a noise". .play() should log: name + " the "+species+" is playing"

Using Object.create()	Using Constructor Functions
<pre>var Animal = { species: '', // comma, not semi-colon makeNoise: function() { console.log("The " + this.species + " makes a noise"); } } var Pet = Object.create(Animal); Pet.name = "default"; Pet.play = function() { console.log(this.name + " the " + this.species + " is playing"); } var dog = Object.create(Pet); dog.name = "fido"; dog.species = "Dog"; dog.makeNoise(); dog.play(); var cat = Object.create(Pet); cat.name = "Mimi"; cat.species = "Cat"; cat.makeNoise(); cat.play();</pre>	<pre>var Animal = function() { this.species = "default"; this.makeNoise = function() { console.log("The " + this.species + " makes a noise"); } } var Pet = Animal; Pet.prototype.name = "default"; Pet.prototype.play = function() { console.log(this.name + " the " + this.species + " is playing"); } var dog = new Pet(); dog.name = "fido"; dog.species = "Dog"; dog.makeNoise(); dog.play(); var cat = new Pet(); cat.name = "Mimi"; cat.species = "Cat"; cat.makeNoise(); cat.play();</pre>



} x not really Pet Subclass -1

Name: _____ StudentID: _____

- 4/10 16. [10 pts] Given the following HTML, write JavaScript with jQuery so that clicking the All button will make all the checkboxes checked, and clicking the None button will make none checked. Hint: you can use the jQuery method: `.prop("checked", boolean)`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Checkboxes</title>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="style.css" type="text/css" />
    <script src="http://code.jquery.com/jquery-2.2.4.min.js"></script>
    <script src="your_code.js"></script>
  </head>
  <body>
    <input type="button" id="all" value="All" />
    <input type="button" id="none" value="None" />
    <table>
      <tr><td><input type="checkbox" name="r1" /></td><td>1st line</td></tr>
      <tr><td><input type="checkbox" name="r2" /></td><td>2nd line</td></tr>
      <tr><td><input type="checkbox" name="r3" /></td><td>3rd line</td></tr>
      <tr><td><input type="checkbox" name="r4" /></td><td>4th line</td></tr>
      <tr><td><input type="checkbox" name="r5" /></td><td>5th line</td></tr>
    </table>
  </body>
</html>
```

All	None
<input checked="" type="checkbox"/>	1st line
<input checked="" type="checkbox"/>	2nd line
<input checked="" type="checkbox"/>	3rd line
<input checked="" type="checkbox"/>	4th line
<input checked="" type="checkbox"/>	5th line

```
$ (function () {
  $("#all").click(function () {
    // $([
    $("input[type:checkbox]").prop("checked", true);
  });
  $("#none").click(function () {
    $("input[type:checkbox]").prop("checked", false);
  });
})
```

// I'm not sure about this \$("input[type:checkbox]"), I think I also
// can query like this \$("table input").prop("checked", true);

Both would have worked correctly

Name: Dinh Tan Luong

StudentID: 985408

17. [10 pts] Write an ASCII based spinner (an animation you show during loading), using JavaScript and jQuery based on the following HTML and CSS.

+10

Your JS should use the revealing module pattern where your module exposes two functions: show() and hide(). Show makes it so that #loading is seen, and changes #spinner every 250 milliseconds, changing the character from - to \ to | to / and back to -. Hide makes it so that #loading is not seen, and stops the animation.

HTML	CSS
<pre>... <div id="loading"> <h2>Loading</h2> <div id="spinner">-</div> </div> </body> </html></pre>	<pre>#loading{ display: none; background-color: white; opacity: 0.75; text-align: center; position: fixed; top: 0px; right: 0px; bottom: 0px; left: 0px; } #loading h2 { font-size: 40pt; margin: 250px 0px 0px; } #spinner { font-size: 60pt; }</pre>

```
var loading = (function () {
  "use strict";
  // var chars = ["-", "\\", "|", "/"];
  var speed = 250;
  var timer; var animation;
  function start() {
    $("#loading").show(); animation = ["-", "\\", "|", "/"];
    timer = setInterval(rotate, speed);
  }
  function stop() {
    clearInterval(timer);
    $("#loading").hide();
  }
  function rotate()
  {
    var current = animation.shift();
    $("#spinner").html(current);
    animation.push(current);
  }
})
```

```
return {  
    "show": start,  
    if "hide": stop  
};  
});
```

Name: Dinh Tan Luong StudentID: 985408

99

W3 Exam

CS472 WAP October 2016 - Prof. Zijlstra

1. [3 pts] What is the difference between a web server and a web container?

- Web server: a program that where it can run web sites (html, ...)
- Web container: is the ear area of a web server where it can run and maintain java servlets and JSPs

+3

2. [3 pts] How long does a session last?

- As long as you set the session-timeout in web.xml.

+3

3. [3 pts] Why shouldn't you bookmark POST requests, what are the problems?

- POST requests have data in the body, not in URL, so you cannot bookmark POST request. Bookmark works with URL only.

+3

4. [3 pts] In what sense are servlets multi-threaded?

- 1 instance of servlet.
- New thread created for every new request.
- service() called on thread.
- All threads. Each thread has own stack.
- All threads have instance variables.

+3

Name: _____ StudentID: _____

5. [3 pts] Explain how Sessions are related to Cookies:

- When a new session created, a new cookie with the key "JSESSIONID" will also be created to store the current session.
- +3 - Server will receive the JSESSIONID cookie and know that if user is logged in or not (for example)

6. [3 pts] Describe the difference between synchronous and asynchronous web communication:

- Synchronous: user have to wait until the request completes to be able to interact with UI elements.
- +3 - Asynchronous: the request will receive data from server in background. User are still able to interact with UI while waiting for the results.

7. [3 pts] What are JSP directives?

- A set of instructions indicates what ^{container} should do when compiled.
- +3 - for example .jsp comments, import, tag lib ...

8. [3 pts] What are the 4 scopes that JSP looks through in order to find a variable

- Application
- Session
- Request
- Page

Name: Dinh Van Luong

StudentID: 985408

9. [3 pts] What does a TLD file do when writing a Custom Tag?

- TLD file specifies the tag name and specifies the tag-class, when tag-body-content for a custom tag.

+3

10. [3 pts] Describe the POST/Redirect/GET pattern, and why it is important:

- POST will process data, then redirect to the result page to get the result (or display result).
- It is important because POST should not return data and you can bookmark the result page.

+3

11. [5 pts] Write code to read the current value of the "amount" cookie. Then create a new cookie called "half" which should hold the value of amount / 2 for the next 30 days.

```
Cookie[] cookies = request.getCookies();
for (int i = 0; i < cookies.length; i++)
{
    if (cookies[i].key == "amount")
    {
        double half = ((double) cookies[i].value) / 2;
        Cookie cookie = new Cookie("half", half.toString());
        cookie.setMaxAge(60 * 60 * 24 * 30);
        response.addCookie(cookie);
    }
}
```

75
4.88

is strong!

Name: _____ StudentID: _____

- 4.5 12. [5 pts] Write a (fully valid, all the needed tags, HTML5) form for requesting the train schedule. It should have fields for departure date, departure time, departure city and arrival city. This form will submit to a Servlet to retrieve a list of trains around that time (next exercise). Do you think the form should use GET or POST? It should use POST

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Train schedule </title>
  </head>
  <body>
    <form action = "Train" method = "POST" >
      <div>
        <label> Departure date </label>
        <input type = "date" name = "date" />
        <label> Departure time </label>
        <input type = "time" name = "time" />
      </div>
      <div>
        <label> Departure City </label>
        <input type = "text" name = "departurecity" />
        <label> De Arrival city </label>
        <input type = "text" name = "arrivalcity" />
      </div>
      <div>
        <input type = "submit" />
      </div>
    </form>
  </body>
</html>
```

Name: Dinh Tan Luong StudentID: 985408

- + 8
13. [8 pts] Write the servlet that the previous exercise will submit to. Your servlet should retrieve a TrainDao object from the Application scope, and use and then use trains.jsp (next exercise) to display the information.

Your servlet does not have to parse the date and time parameters, the dao will do so, but it should check that all the parameters exist and are not empty.

```
public interface TrainDao {  
    List<Trip> getTrips(String date, String time, String departure, String arrival);  
}  
  
public class Trip {  
    private String departureTime;  
    private String arrivalTime;  
    private int transfers;  
    private String duration;  
  
    // Getters and Setters not shown  
}  
  
@WebServlet ("/Train")  
public class Train extends HttpServlet {  
  
    protected void doPost (HttpServletRequest request, HttpServletResponse response)  
    {  
        Train Dao dao = (Train Dao) getServletContext ().getattribute ("traindao");  
        String date = request.getParameter ("date");  
        String time = request.getParameter ("time");  
        String departure = request.getParameter ("departure");  
        String arrival = request.getParameter ("arrival");  
  
        if (date == null || time == null || departure == null || arrival == null)  
        {  
            response.sendRedirect ("/");  
            return;  
        }  
  
        List <Trip> trips = dao.getTrips (date, time, departure, arrival);  
        HttpSession session = request.getSession ();  
        session.setAttribute ("trips", trips);  
        response.sendRedirect ("trains.jsp");  
    }  
}
```

Name: _____ StudentID: _____

14. [8 pts] Write the trains.jsp page to display all the possible trips, be sure to include columns for: departure time, transfers, arrival time, and duration (see Trip class on previous page). To save time you do not have to write all the HTML tags – just what would be inside <body>.

8

```
<table>
  <tr>
    <th> Departure Time </th>
    <th> # transfers </th>
    <th> Arrival time </th>
    <th> duration </th>
  </tr>
  <c:forEach item="#${trips}" var="trip">
    <tr>
      <td> ${trip.departureTime} </td>
      <td> ${trip.transfers} </td>
      <td> ${trip.arrivalTime} </td>
      <td> ${trip.duration} </td>
    </tr>
  </c:forEach>
</table>
```

15. [7 pts] We're going to make a chat application. To start write a servlet that receives the requests for the latest messages, the client (AJAX in the browser) will send the ID of the last seen message. You should get the ChatDao object from the Application context and use `List<String> chatDao.getMsgsAfter(int msgId)`. Next question we'll write the JSP.

6.75

```
@WebServlet("/Messages")
public class Messages extends HttpServlet {
  protected void doGet(HttpServletRequest request, HttpServletResponse response) {
    ChatDao dao = (ChatDao) getServletContext().getAttribute("chatdao");
    int msgId = (int) request.getParameter("msgid");
    List<Message> messages = dao.getMsgsAfter(msgId);
    request.setAttribute("msgs", messages);
    RequestDispatcher view = request.getRequestDispatcher("message.jsp");
    view.forward(request, response);
    request.setAttribute("view", view);
    view.forward(request, response);
  }
}
```

Name: Dinh Tan Luong StudentID: 985408

16. [7 pts] Write the JSP that outputs the JSON to send the messages ✓

```
{<c:forEach begin = "0" end = "${msgs.length}" var = "i">
    {
        "id": ${msgs[i].id},
        "text": "<c:out value = '${msgs[i].text}' />"

        <c:if test = "${i < msgs.length - 1}">
            <c:out value = "," /> ✓
        </c:if>
    }
}</c:forEach>
```

17. [7 pts] write a servlet that receives a message from the user. Your servlet should retrieve
the user's name from the session and concatenate it in front of the message text. Then use
void chatDao.insertMsg(String msg) to add the message to the database. This servlet
should not output any response, or forward, or redirect, the client expects nothing back.

```
@WebServlet("/MessageController")
public class MessageController extends HttpServlet {
    protected doPost(HttpServletRequest request, HttpServletResponse response) {
        ChatDao chatDao = getServletContext().getAttribute("chatdao");
        String username = request.getSession().getAttribute("username");
        String text = request.getParameter("text");
        String msg = username + ":" + text;
        chatDao.insertMsg(msg);
    }
}
```

Name: Dinh Tan Luong StudentID: 985408

- 7 18. [7 pts] Write a Filter that sits in front of the Servlets and JSP you wrote in the previous exercises. It should redirect to the login page if no name is found in the session.

```
@ WebFilter (name = "chatfilter", urlPatterns = "/Messages", "/messages.jsp")  
public class chatfilter extends HttpServletFilter {  
  
    protected void doFilter (ServletRequest request, ServletResponse response,  
                           FilterChain chain) {  
  
        HttpSession session = ((HttpServletRequest) request).getSession();  
        String username = (String) session.getAttribute ("username");  
        if (username == null || username.isEmpty ())  
        {  
            ((HttpServletResponse) response).sendRedirect ("login.jsp");  
            return;  
        }  
        chain.doFilter (request, response);  
    }  
}
```

19. [10 pts] Write the JavaScript for the chat application. When the page loads there is a hidden field (#lastMsgId) that holds the id of the last message the server had.

10

Once every second your page should send an AJAX GET request for new messages (sending the id of the last seen message). When the data comes in it should append each message to the div #msgs, and of course keep track of the ID of the last message as the new 'lastId'

Also when the user presses the send button (#send), you should AJAX POST what was typed in the text input (#msg) to the server.

```
$ (function () {  
  
    var msgid = $("#lastMsgId").val ();  
    var timer = setInterval (getMessages, 1000);  
  
    $("#send").click (function () {  
        var msg = $("#msg").val ();  
        $.post ("MessageController", {"text": msg});  
    });  
  
    function getMessages ()  
    {  
        var msgid = $("#lastMsgId").val ();  
        $.get ("Messages", {"msgid": msgid}, dataType: "json");  
    }  
});
```

Name: Dinh Tan Luong

StudentID: 985408

19) (continue)

```
function getMessages() {
    var msgid = $('#lastMsgId').val();
    $.ajax({ url: "Messages",
        data: { "msgid": msgid },
        dataType: "json"
    }).done(function(data) {
        $.each(data, function(index, item) {
            var div = $("<div>" + item.text + "</div>");
            $("#msgs").append(div);
            $("#lastMsgId").val(item.id);
        });
    });
}
```

Review - Event Handling

DEFINING EVENT HANDLERS AND THIS

How would you attach the function myFunc as click event handler to all divs using jQuery?

```
$(‘div’).on(‘click’, myFunc);
```

How would you trigger a click event for all divs?

```
$(‘div’).click();
```

Or

```
$(‘div’).each(function(indx, elem) {  
    $(elem).click();  
});
```

What is an event object? What are clientX, pageX, and screenX for mouse event objects?

An event object is an object that will be passed to an event handler by the browser if the event handler has a parameter. It contains information about the event, such as clientX, pageX, and screenX are event object properties for mouse-related events.

What does the keyword 'this' refer to in an event handler code?

‘this’ refers to the element that registered the event handler – e.g., a button registers the on click handler for that button.

EVENT BUBBLING

Which event fires first if there are multiple event handlers assigned to the same element?

Generally, event handler assigned first should fire first. However, there is no guarantee that this will be enforced and is the case at all times. Therefore, we should refrain from structuring our code in such event handlers presumptuous of such ordering.

Is event bubbling for the handlers on the same element or different elements?

Event handling concerns with different elements in containment hierarchy.

What element relationships do events bubble to?

Containment (child-parent relationship) across ancestry hierarchy.

What is the difference of stopPropagation and stopImmediatePropagation and preventDefault?

stopPropagation() method of event object stops further bubbling of an event.

stopImmediatePropagation() method of event object stops bubbling but also stops other event handlers registered on the target element for the same event. **preventDefault()** method of event prohibits the default behavior of related elements. Default behavior of form is to submit whatever input is collected to respective processing server-side URL explicitly stated in its ‘action’ attribute or

deduced to be the same page when none is specified. Default behavior of clicking on an anchor element is to browse to the specified URL.

What happens when an event handler returns false?

An event handling returning 'false' is effectively doing the following: 1) stopping the event from bubbling up the ancestral hierarchy. 2) Stopping any default behavior instilled for the element from happening. In essence, return 'false' from an event handling is equivalent to calling **`evt.stopPropagation()`** and **`evt.preventDefault()`** on the associated event object with in the event handler code.

What is event delegation?

This is a technique for associating an event for element(s) with the container(ancestor) element. The event handler at the ancestor might need to inspect the event object against being the correct target to apply its logic. This is handy when the elements are dynamically created because we don't have to explicitly bind them to event handlers every time.

EVENT LOOP

What is the call stack?

This is where currently executing functions and their local variables are stored during execution.

What is the JavaScript callback-task queue and how is it involved in the event loop?

This is where callback functions (such as event-handling callbacks and timer events) are queued until the event loop takes them to call stack for execution. They are put in the queue when they fire, i.e., when that event happens.

When do timer events fire?

Timer events fire when their delay time elapses. When they are fired, they will be put into the callback-task queue awaiting for their execution turn; that is when the call stack is free and they are at the forefront of the callback-task queue.

Review Questions- jQuery

SELECTION

What is the jQuery way to do `window.onload = function () ... ?`

First Way: `$(document).ready(function() {}); //jquery function`

Second Way: `$(function() {}); //another jquery shortcut`

Write a jQuery selector to find all paragraphs.

`$('.p')`

Write a jQuery selector to find all list items inside paragraphs that have the class `.donkey` .

`$('.p li.donkey');`

What are the jQuery symbols used in selecting: by identifier, by class, children descendants, any descendent, by collections of selectors, by HTML element, by attribute, by attribute value?

- Selecting by Identifier: '#'
- Selecting by Class: '.'
- Selecting Children Descendants: '>' (* after enclosing element)
- Selecting Any Descendant: '' (*space after enclosing element)
- Selecting by Collection of Selectors: Comma-separated selectors e.g. `$('.p, div, div.warnings')`
- Selecting by HTML element: `($('elementName')` (* here `elementName` denotes tag name for the element: e.g. p, div, span, etc)
- Selecting by Attribute: `($('element[attributeName*="value"]'))` /* For attributes checked for just presence; such as 'required' attribute in input element, value can be unspecified. For instance, selecting radio button in a form can be specified as follows: `($('input[type="radio"]'))`

What does the jQuery filter '`:even`' do? Give an example.

This selects the even-indexed elements in the selected set of elements. `($('.tr:even'));`

What is the difference of the 'jQuery function' and 'a jQuery object' ?

'jQuery Function' refers to the global jQuery function that is normally aliased as \$ for brevity. Where as, 'jQuery object' the object returned by the jQuery function that often represents a group of elements

What does it mean to use \$ as a wrapper? What is the purpose?

Turns whatever is enclosed to 'jQuery Object'; which opens up to various useful functions.

What is the jQuery syntax for searching a subtree of elements?

`$(selector, $elem); // Where $elem denotes where the root of the subtree`

TRAVERSAL

Give an example of finding the parent nodes of all div elements.

```
$('.div').parent();
```

Give an example of finding the children nodes of all div elements.

```
$('.div').children();
```

Give an example of finding the descendant nodes of all div elements.

```
$('.div').children('*');
```

Give an example of finding the sibling nodes of all div elements.

```
$('.div').siblings()
```

Give an example of finding the third sibling nodes of all div elements.

```
$('.div').siblings().eq(2);
```

What is the difference of \$("p").each(myfunction) and \$.each(myArray, myfunction)?

jQuery's each() function(the former) is used to loop through each element of the target jQuery object. In the latter version of each utility function, the collection to loop through is given as the first argument.

NODE MANIPULATION

Show how to add class .puppy to all divs.

```
$('.div').addClass('puppy');
```

Show how to remove class .puppy from all divs.

```
$('.div').removeClass('puppy');
```

Show how to remove class .puppy from any div that has it and add .puppy to any div that does not have it.

```
$( "div.puppy" ).removeClass( "puppy" );
```

```
$( "div:not(.puppy)" ).addClass( "puppy" );
```

What is the syntax for getting a style property in jQuery?

```
$(selector).css(property);
```

What is the syntax for setting a style property in jQuery?

```
$(someSelector).css(someProperty, someValue);
```

What is unobtrusive styling? What is the implication of this for setting style properties in jQuery versus setting classes?

JavaScript/jQuery should change styling by adding/removing defined classes to elements other than directly manipulate style attributes in their respective DOM.

What is the important difference in how jQuery setter and getters work?

One difference is the arguments such functions take. In most cases, getter methods accept the property to be retrieved and return that value. Setter methods, normally, take on two arguments where one is the property concerned and the other value to be set for it. The setter methods return same processes jQuery object that enable method chaining. The intent of getter methods, quite obviously, retrieving existing value for the specified property. Where as, jQuery setters have intent of changing previous property value with new

one.

Explain why the following will not work (as expected):

```
$("#div").css("top", parseInt($("#div").css("top")) + 100 + "px");
```

Describe what is returned by each of the following and the implications of this for 'chaining':

```
$("#myid");
```

jQuery object is returned; making it ready for further method chaining.

```
$("#myid").css("margin-left");
```

the return type is the value for margin left which is some metric together with a measurement(px, etc). This return type does not afford method chaining.

```
$("#myid").addClass("special");
```

jQuery object with the newly added class will be returned; which allows further method chaining.

What is the difference of .html() and .text()? Which should be favored to avoid 'HTML hacking'?

The text() method sets or returns the text content of the selected elements. When this method is used to return content, it returns the text content of all matched elements; with HTML markup removed. The html() method, on the other hand, sets or returns the content (innerHTML) of the selected elements; and do so with out escaping html tags. .text() should be preferred to avoid 'HTML hacking' as it, by default, escapes html tags.

TREE MANIPULATION

How would you create a new div node with id = 'BigRed' and class = fancy in jQuery? (use the property object syntax)

```
let newEl= $('div').addClass('fancy').attr('id','BigRed');
```

Will this appear in the DOM when you create it?

No, it is appended for last child, prepend for first child in the context of existing DOM element.

What is necessary to add it to the DOM?

append it as last child or prepend it as first child for existing DOM element. Example,

```
$('#myId').append(newEl); // newEl being the object created early on
```

What are the 4 jQuery function signatures? Explain what the argument types are for each and what each returns.

1. `$(function);`
2. `'selector', [context];`
3. `'elements';`
4. `'<html>', [properties];`
5. `('css selector', jQuery);`

Review questions for Servlets and Containers

SERVLETS, SERVERS, WEB CONTAINERS

What is the difference of a web server and a web container?

- Web servers are only capable of serving static resources (CSS, HTML, Video, Image, etc); whereas web containers serve dynamic content that depend on user input, characteristics, etc.
- Web containers are usually affiliated with a specific technology(Java, PHP, etc) that makes it possible to generate dynamic content in response to user requests. Web servers are, in most cases, built to work with multitude of technology families.
- Webserver has direct interface with HTTP clients; but web containers,usually, serve client request with mediation (orchestration) from the webserver.

What is a servlet?

A servlet is a server-side java code capable of responding with dynamic content for HTTP requests.

How do web servers and web containers interact with servlets?

Client requests which are identified to be URLs for servlets are handed over by web server to web container. The Web container would find the respective servlet, create a thread of execution for it, encapsulate request and response objects, handover to the service method of the servlet (which decided what specific method to execute depending on the HTTP request method), grab output from execution of the servlet(from processing done in it) and send back to the web server. The web server would, then, respond to the original user request with the dynamically generated resource from the container.

WEB CONTAINER SUPPORT SERVICES

Who creates request objects?

The web container right before the invocation of the service method of targeted servlet.

Who creates response objects?

The web container right before the invocation of the service method of targeted servlet.

How do HTTP messages relate to the doGet and doPost methods in a servlet?

The client HTTP requests made will be inspected by the web container to decide the METHOD of request. This helps the service() method of the servlet which specific method of do*() to hand over it to.

LIFECYCLE

What are the states in the servlet lifecycle?

- initialization.* that occurs after a servlet is constructed and prior to providing service. useful to set servlet-wide initializations.
- service.* that constitutes the duration where client requests are accommodated with respective threads to cater for responses.
- destroy.* The juncture in time that occurs when the servlet is put off by the container.

Who calls init and when?

init() is invoked by web container after a servlet is first constructed but before it starts serving client requests.

Who calls service() and when?

service() method is called by web container for every request of the servlet.

Who calls doGet and when?

doGet() will be called from service() method of a servlet by web container from the thread spawned for a client request to the servlet using HTTP request method.

Which of init, service, and doGet should you override?

service() method should not be overridden. init() could be overridden if such actions are justified. doGet() should be overridden for the servlet if we want the servlet respond to GET HTTP request.

In what sense are servlets multi-threaded?

One instance of the servlet is created by the container, and then a new thread is created for each request.

What are the implications of this for servlet instance variables?

The servlet and its instance fields are not locked by specific code at any given time but open for competition by existent threads. Stated differently, various threads of the servlet have access to and potentially alter the instance fields; leaving their state inconsistent. Generally, developers should not use instance fields in servlets because of the potential race conditions.

W3D4 State Management Review Questions

GET/POST FORWARD/REDIRECT

Can post requests be bookmarked? What are the problems?

Yes they can in a sense, but as get requests and without any query parameters. Those browsers who do, strip of the body of the 'post' request; turning it to 'get' request. The underlying problem is the fact that 'post' requests are not idempotent; which entail a server side update and processing to take over. If they were to be bookmarked properly they would result in those server updates and processing to take place; which have ramifications on the validity and consistency of server-side information.

What is the purpose of request dispatching?

Enabling one servlet delegate part of or all responsibility with other parts of the web application (other servlets and/or JSPs). This is useful to attain separation of concern among modules in our applications. This is the very mechanism that allows us realize the MVC pattern in servlet-based applications where the servlet acts as controller- interpreting user action to requests in model and furnishing information for use by view(JSP) after it delegates using request dispatching.

What is the difference between redirect and request dispatch?

With request dispatch, the request is handed over to other delegated processing servlet/JSP with out awareness of and entailing further communication overhead. The user shall see no change in address bar of the browser s/he is using. The delegation almost always happen in the same application scope; which enable state management to succinctly operate.

Redirecting, on the other hand, returns a 300's status line with associated redict URL for the requesting client where further communication using the URL proposed gets the initial task performed. The user shall see a change in the browser's address bar. The redirected URL might not always be with in the same application scope; therefore, state management is not possible.

ATTRIBUTES

What is an attribute?

Is an object in web container that promotes integration and communication between components of the application.

What are scopes of attributes?

Request scope, Application/Context scope, and Session Scope.

What is the difference between attributes and parameters?

Parameters data sent from HTTP user collected in form elements or as part of query string. They are name/value pairs of string types. These are sent from user users and accessed in the server. Attributes are web container managed name/value pair objects where the name is a string and the value an object. They are used for communication and coordination of server side components.

Which scopes are thread safe?

Only Request scope is thread safe.

What are dangers of using attributes?

Depending on their scope, they might not be thread safe.

SESSIONS AND COOKIES

What does it mean to say that http is stateless? Give an example of a stateful protocol.

Stateless means that every request is treated as new by the server. IP is other stateless protocol that sends packets and forgets about them.

What is a cookie, and how long does a cookie last?

Cookie is a little piece of data(name/value pair) exchanged between client and server. Cookies for which maxAge() is not specified last as long as the user session; that is until the user closes the browser. setMaxAge() method can be used to explicitly set how long cookies should last in terms of seconds. In such circumstances, it should last as long as the specified value. If the user disables browser cookie support, they might not exist to start with. User action to clear cookies in the browser setting might prematurely discard cookies before their preset expiry time.

How are cookies involved in session management?

The container uses temporary cookies underlying its session management.

How long does a session last?

The application-level expiry time of session could be done in WEB.xml using <session-timeout> setting. When sessions, with no individual expiry time set sit idle past this time set in this setting, they expire automatically. Sessions can have specific expiry time set using .setMaxInactiveInterval() method. For such session instances, past the time set for them denote their expiry time. Any session can prematurely forced to expire using the method .invalidate(); which automatically expires it.

What is a hidden field and how can it be used to maintain state information?

Hidden field is an input element of type 'hidden' that is set to carry special values that serve maintain conversational states. They are invisible to the user but accessible to servlet and JSPs.

Give 5 different methods for maintaining state information (count each attribute scope as one method)

1. Using the HttpServletRequest attributes as part of the container managed state management
2. Using the HttpSession attributes as part of the container managed state management
3. Using the ServletContext attributes as part of the container managed state management
4. Using cookies
5. Using hidden input fields

Suppose you create a web application and then create a servlet with the doGet and doPost from the maintaining state demo for today's lesson, and that you map the url /stateDemo to the servlet.

Now if you were to run the web app and then enter <http://localhost:8080/stateDemo> in the browser url bar. What will appear in the browser? (Do not really implement it, answer it based on your understanding from reading the code.)

What will appear in the browser after pressing the ‘Click me’ button? (for the /stateDemo question)

What will the doPost put in the output window?