Name: NAVIN PAUDEL StudentID: 611010

77.9

Final Exam

CS472 WAP July 2020 – Prof. Zijlstra

Theory Questions

a. [3 pts] Give 3 ways that you can stop an event from bubbling with jQuery

2.9

① stopPropagation() (or) event.stopPropagation
② stopImmediate() (or) event.stopImmediate.
③ event.preventDefault() ✗
④ Returning false in jQuery. ✓

b. [3 pts] Give an advantage of configuring a web application in xml with web.xml, and also give an advantage of configuring a web application with annotaitons.

3

→ By configuring a web application in xml with web.xml, we can make link or can call a servlet i.e one or more servlet by giving a servlet-name, servlet-class and map it using url-pattern.
→ with annotations, we don't need ~~web.xml~~ to configure web.xml and can directly link by using @webservlet("/test").

c. [3 pts] Describe the difference between GET and POST

→ GET means to receive a data from the Servlet through server.
→ POST means to send data to the servlet through server.

3

d. [3 pts] Describe what an HTTP redirect is, and how you can do one in a Java Servlet

→ Http redirect is external, and it tells browser to go a different URL (make new request for new URL).
→ It is used to call another servlet so using HttpServlet ~~request~~ response object we call another servlet. eg. HttpServlet ~~request~~ response response. then ~~request~~ response.sendRedirect("sq") → here sq is mapping to another servlet.

3

e. [3 pts] Describe what the differences and similarities are between cookies and sessions

→ A session cookie is a unique identifier that the server gives to a client so that it can identify the client. Although the server can store a lot of things for the client all the cookie ever hold is sessionid.
→ By using HTTPSession, session ~~we~~ can setAttribute so that it can be used until session is there.
→ using cookie, we can add cookies and use same cookies to access the data from the servlet.

3

f. [3 pts] When are JSP pages translated to Java?

→ Path followed by JSP are ① Compilation ② Initialization ③ Execution ④ cleanup. so first parsing JSP and turning JSP into servlet i.e Java and then ~~the~~ compiling the servlet is done.

2

on first request

g. [3 pts] Describe what is meant with a "model 1" architecture

→ Model 1 Architecture means Client sends a request to server and server sends to servlet and servlet sends data to jsp and jsp will put that data in its model and send it to client.

0

that's model 2

Name: NAVIN BAUDEL StudentID: 61101 0

h. [3 pts] What does MVC stand for, and how do you implement it with Java Web Technologies?

2 → MVC stands for modify view cancel (cancel means delete).

→ MVC is implement with Java web Technologies using JSTL X model Pojo view JSP control Servlet

i. [3 pts] Describe what the Same Origin Policy is in JavaScript:

→ Same origin policy means if we make a object using class or if we add properties or method using prototype then both are same.

0

j. [3 pts] What is meant with Callback Hell?

→ main disadvantage of callback approach occurs with multiple chained asynchronous tasks. It requires us to define callback function within callback functions within callback functions called Callback hell.

3

Code Questions

```
public class CartItem {
    private String name;
    private double price;
    private int quantity;

    public CartItem(String n,
            double p, int q) {
        this.name = n;
        this.price = p;
        this.quantity = q;
    }
    // all getters and setters
}
```

1. **Shopping Cart Question.** The session will store a List<CartItem> as being "The shopping cart"; see the code below – Assume getters/setters exist even though they are not shown here.
   - [7 pts] **Write a servlet mapped to "/addToCart"** and can receive POST requests that contain the following parameters: productName, productPrice, quantity. Your servlet should create a CartItem based on the input. Make a List<CartItem> if it's not in the session, and update the session. Once the session is updated it should redirect to "viewCart"

6

   - [3 pts] **Write a servlet mapped to "/viewCart"** and can receive a GET request. The servlet adds a message to request that says "Here is your cart" if a cart exists in the session, or "Your cart is empty" if no cart exists in the session. It then forwards to cart.jsp

2.5

   - [10 pts]**Write a JSP page** called cart.jsp that displays the message from the servlet and then displays the contents of the cart. Be sure to write correct HTML5 in your JSP

9.5

```
@WebServlet("/addToCart")
Public class addToCart extends HttpServlet{
  @Override.
Public void doPost(HttpServletRequest request, HttpServletResponse
                                                        response)
      Throws HttpException, IoException{
  List<CartItem> items = new ArrayList<>();
String productName = request.getParameter("productName");
int productPrice = Integer.parseFloat(request.getParameter(
                                "productPrice"));
```

Name: NAVIN PAUDEL StudentID: 611010

extra space for answering q1

```
int quantity = Integer.parseInt(request.getParameter("quantity"));
        CartItem item = new CartItem();
            item.name = productName;
            item.Price = productPrice;
            item.quantity = product quantity;

        items.add(item);
        HttpSession session = request.getSession();
        session.setAttribute = ("items", items);
        response.sendRedirect("viewCart");

    }
    @webServlet("/viewCart")
    Public class viewCart extends HttpServlet {
      @ override
      Public void doGet(HttpServletRequest request, HttpServlet Response
              response) throws HttpException, IoException {
        List<CartItem> items1 = new ArrayList<>();
        HttpSession session = request.getSession();
items1 = session.getAttribute("items");
        itemsAdd = "Here is your cart";
      request.setAttribute("items1", items1)
      request.getRequestDispatcher("Cart.jsp").forward(request, response);

    }
    <%@ taglib prefix='c' uri="http://java/un.com/jsp/jstl/core %>
    <html lang="en">
    <head>
        <meta charset: "UTF-8">
        <title> The shopping cart</title>


    </head>
    <body>
      <c:forEach var="i" value="items1">
          <c:set    var="j" value="${i}">
            <p> message: <c:out value="${j[3]}"/></p>
            <p> ProductName: <c:out value="${j[0]}"/></p>
            <p> Product Price: <c:out value="${j[1]}"/></p>
            <p> quantity: <c:out value="${j[2]}"/></p>
```

fields are private

What if session already has "items"?

Cannot use square brackets to get name price quantity

Name: NAVIN PAUDEL StudentID: 611010

</c:forEach>

</body>
</html>

2. **AJAX & JSON.**
   - [10 pts] **Make a page that looks like the screenshot** Write HTML5 and use CSS for the layout. The dropdown should have the options "walking", "riding a bike", "driving a car".
   - [10 pts] **Write JS with jQuery and Ajax** to GET Advantages / disadvantages from "/transport" when the user selects a value in the dropdown. It should pass the value of the dropdown as a data parameter called type so that it knows which advantages / disadvantes to get. The data will come back as **JSON** (see example below). Update the advantages area to show the advantages (as an unordered list), and update the disadvantages area to show the disadvantages (as an unordered list).

```
{"advantages":["healthy", "cheap", "always available"],
"disadvantages": ["slow", "boring", "tiring"] }
```


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href="Quez2.css" rel="stylesheet">
  <script src="https://ajax.googleapis.com/ajax/
    libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="Quez2.js" rel="stylesheet">
  <title>Quez2 Solution</title>
</head>
<body>
  <select class="list">
    <option class="walk">walking</option>
    <option class="ride">riding a bike</option>
    <option class="drive">Driving a Car</option>
  </select>
  <td>
    <th class="advantage">Advantages</th>
    <th class="disad">Disadvantages</th>
  </td>
```

Quez2.css

```
.list { text-align: center }
td th { font-weight: bold }
```

will not look like

Name: NAUIN PAUDEL StudentID: 61/010

extra space for answering q2

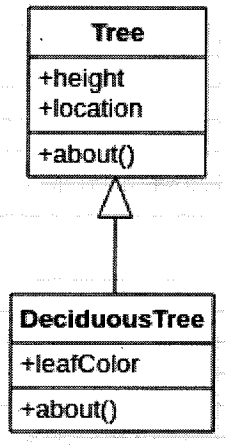
```
</body>
</html>
```

Quest. js

```
$(document).ready(function(){
    var $list = $("#list");
    var $walk = $(".walk");
    var $ride = $(".ride");
    var $drive = $(".drive");
    var $adv = $(".advantage");
    var $dis = $(".disad");
  $.ajax({ X needs to fire an event
    type: 'GET',
    url: 'http://jsonplaceholder.typicode.com/posts',
    success: function(list){
      $.each(list, function(list){
        if($walk){
          $adv.append('<li>' list.advantages[0] '</li>')
          $dis.append('<li>' list.disavantages[0] '</li>')
        }
        if($ride){
          $adv.append('<li>' list.advantages[1] '</li>');
          $dis.append('<li>' list.disadvantages[1] '</li>');
        }
        if($drive){
          $adv.append('<li>' list.advantages[2] '</li>');
          $dis.append('<li>' list.disadvantages[2] '</li>');
        }
      });
    },
    error: function(){
      alert("Couldnot load");
    }
  });
```

Name: NAVIN PAUDEL StudentID: 611810

3. [10 pts] **Implement JS classes** as shown in diagram. The about() method on tree should return a String "Height: " + height + " Location: " + location, The about() method on DeciduousTree should return a String that is based on the string from Tree, but also adds " Leaf Color: " + leafColor. Be sure to use super in both the constructor of the subclass and the about method of the subclass!



9

```
class Tree{
    constructor(height, location){
        this.height=height;
        this.location = location;
    }
    about(){
        return("Height:" + height + "Location:" + location)
    }
}
class DeciduousTree extends Tree{
    constructor(height, location, leafColor){
        super(height, location);
        this.leafColor = leafColor;
    }
    about(){ super
        return("Height:" + this.height + "Location:" + this.location +
               "LeafColor:" + this.leafColor)
    }
}
```

4. **Write jQuery code** that adds an the following two event handlers to all input elements of type text on the page. Assume that all input type text elements on the page will have a width set on them in pixels, but some have a different width than others.
   - [3 pts] The code for attaching your event handlers should run once the page has loaded
   - [6 pts] When the input element receives focus it should double its width
   - [6 pts] When the input element loses focus (blur) it should halve its width

3
4.5
4.5

```
$(document).ready(function(){
    $("input").focus(function(){
        $("input").css("width", "80px");
    }
    $("input").blur(function(){
        $("input").css("width", "20px");
    }
}
```