

Sample CS 142 Midterm Examination

Winter Quarter 2016

You have 1.5 hours (90 minutes) for this examination; the number of points for each question indicates roughly how many minutes you should spend on that question. Make sure you print your name and sign the Honor Code below. During the examination you may consult two double-sided pages of notes; all other sources of information, including laptops, cell phones, etc. are prohibited.

I acknowledge and accept the Stanford University Honor Code. I have neither given nor received aid in answering the questions on this examination.

(Signature)

(Print your name, legibly!)

(SUID - stanford email account for grading database key)

Problem	#1	#2	#3	#4	#5	#6	#7	#8	#9	Total
Score										
Max	16	18	8	10	9	8	8	7	6	90

~~Problem #1 (16 points)~~

In class we described the structuring of web applications in **model**, **view**, and **controllers**. For each of the pieces of an web application implementation listed below, state which parts (model, view, or controller) would the piece be most associated with. Provide a brief justification for all your answers.

- A. CSS rules
- B. JavaScript code
- C. HTML
- D. Angular Directive
- E. Angular Service
- F. Database data
- G. DOM event handlers
- H. Animations

~~Problem #2 (18 total points)~~

While submitting your class assignment at the last minute before the deadline you discover you have the view template and a screenshot of the view running but appear to have lost the Angular controller for the view. Your job is to create a controller that given the view template would plausibly generate what is seen in the screenshot. **Write your answer on the following page.**

----- view template -----

```
<div ng-controller="MyController">
  <div>
    <label>Name:</label>
    <input type="text" ng-model="person.name" placeholder="Enter a
name">
    <h1>{{greeting}} {{person.name}}!</h1>
    <ol>
      <li ng-repeat="c in favColors">
        <span>{{c.name}}'s favorite color is {{c.color}}</span>
      </li>
    </ol>
    <button ng-if="buttonShow" ng-click="buttonClicked()">
      Button #1 (has been clicked {{buttonClicks}} times)
    </button>
    <p ng-show="buttonClicks % 2">
      Button #1 was clicked an ODD number of times
    </p>
    <p ng-show="!(buttonClicks % 2)">
      Button #1 was clicked an EVEN number of times
    </p>
  </div>
</div>
```

----- screenshot -----

Name:

Hello Jack!

1. Fred's favorite color is Blue
2. Sue's favorite color is Red

Button #1 (has been clicked 5 times)

Button #1 was clicked an ODD number of times

... Problem 2 continued

Implement the controller for the view template on the previous page. (16 points)

```
angular.module("cs142App", [])  
    .controller("MyController", function($scope) {
```

```
});
```

... Problem 2 continued

(2 points)

The view template in this problem uses both an `ng-if` attribute on the button element and an `ng-show` attribute on the odd/even message paragraph elements. Given the usage in the template and controller, do you agree with the use of `ng-show` instead of `ng-if` for the paragraph elements? Provide an argument as to why or why not.

Problem #3a (4 points)

You direct a browser to a URL `http://www.example.com/initial-page/index.html` which replies with a web page with the following links:

```
<a href="/starts/with/a/slash">Link A</a>
```

```
<a href="no/starting/slash">Link B</a>
```

Provide the full URL that would be navigated to when you click on each of the links:

I. Link A

II. Link B

Problem #3b (4 points)

What does the web server typically do with the fragment part of the URL? What does the browser do with it?

Problem #4a (5 points)

Describe what is meant by a **CSS breakpoint**. Include a description of how breakpoints are implemented and give an example of how they are used.

Problem #4b (5 points)

The `a` tag in HTML in addition to the attribute `href=` supports an attribute named `target=`. For example a link to the Google search page could be written as:

```
<a target="_blank" href="http://www.google.com">Google</a>
```

The `target="_blank"` causes the browser to open a new window for the page being fetched by the link rather than the default (`target="_self"`) which switches the current window to the link.

Although traditionally links that open new windows were frowned upon for annoying users, single page applications tend to use them for all external links. Give a reason why this is the case.

Problem #5 (9 points)

Consider the following fragments of HTML and JavaScript:

----- HTML -----

```
<div id="a">A
  <div id="b">B
    <div id="d">D</div>
  </div>
  <div id="c">C</div>
</div>
```

----- JavaScript -----

```
function callback(event) {
  console.log(event.target.id + ' ' + event.currentTarget.id)
}
document.getElementById('a').addEventListener("click", callback, false);
document.getElementById('b').addEventListener("click", callback, false);
document.getElementById('c').addEventListener("click", callback, false);
document.getElementById('d').addEventListener("click", callback, false);
```

- I. What would be printed out if the user clicked on the letter 'D'?

- II. If the order of the calls to `addEventListener` were reversed so we instead registered the listeners in the div id order of 'd', 'c', 'b', 'a', what would be printed out if the user clicked on the letter 'D'?

- III. If we change the JavaScript so that the third argument to each `addEventListener()` call is `true` (i.e. register them as capture rather than bubble listeners) and keep the original order from Part I, what would be printed out if we clicked on the letter 'C'?

Problem #6 (8 points)

Consider the following JavaScript:

```
function Jeep() {  
    this.organization = "DaimlerChrysler";  
}  
Jeep.prototype.color = "blue";  
  
var blue = new Jeep();  
  
console.log(blue.organization);    // Line A  
console.log(blue.color);          // Line B  
  
var red = new Jeep();  
red.color = "red";  
  
console.log(red.organization);     // Line C  
console.log(red.color);           // Line D  
  
Jeep.organization = "Chrysler";  
Jeep.prototype.color = "green";  
  
console.log(blue.organization);    // Line E  
console.log(blue.color);          // Line F  
console.log(red.organization);     // Line G  
console.log(red.color);           // Line H
```

In the table below list what is printed at each console.log statement:

console.log statement	What is printed by the console.log
Line A	
Line B	
Line C	
Line D	
Line E	
Line F	
Line G	
Line H	

Problem #7 (8 points)

- I. What mechanism is most commonly used to tie the styling information in CSS files to the HTML elements they style?
- II. If your CSS doesn't set the width and height of an element, how is the width and height of that element determined?
- III. Explain the difference between a CSS position of `fixed` and `absolute`.
- IV. What is the difference between marking a element with the CSS `"display: none;"` and marking the element `"visibility: hidden;"`?

~~Problem #8 (7 points)~~

- I. Web applications are frequently constructed as a distributed system utilizing a multitiered architecture with three tiers. List and provide a brief description of the three tiers.

- ~~II. AngularJS contains a number of concepts including templates, controllers, scopes, directives, and services. For each of the following parts of a web application state which AngularJS concepts from the list above would be most relevant for the part's implement. Provide a brief justification with your answers.~~

~~A. A JavaScript library for communicating with the web server to get data models.~~

B. A help widget that is placed in the same spot on every view of the web application. The widget displays helpful information fetched from the web server.

Problem 9 (6 points)

- I. Some JavaScript coding standards require that `var` statements in functions are always placed at the start of the function, rather than close to where the declared variable is used (as is typically done in C and Java programs). Give a compelling reason for this recommendation.

- II. What does the following bit of JavaScript print out?

```
var a = [1,2,,4,5];  
console.log([a[4], a[2], a[5]]);
```

- III. What does the following bit of JavaScript print out?

```
var x = 1;  
(function (x) {  
    console.log(x);  
    x = 3;  
})(2);  
console.log(x);
```