

Sample CS 142 Midterm Examination

Fall Quarter 2017

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!)

_____@stanford.edu
(Stanford email account for grading database key)

Problem	#1	#2	#3	#4	#5	#6	#7	Total
Score								
Max	13	8	18	12	15	12	12	90

Problem #1 (13 points)

1A (8 points)

Review the following HTML document, which contains inline CSS. Below the HTML document are several different screenshots of an HTML document rendered in the browser. Circle the letter corresponding to the **correct** rendering of the HTML document.

```
<html>
<head>
  <title>CS142</title>
  <style type="text/css">
    body {
      border: 5px dotted black;
      font-size: 10;
    }
    div {
      height: 100px;
      border: 5px solid black;
    }
    .div2 {
      visibility: hidden;
    }
    .div3 {
      font-size: 20;
    }
  </style>
</head>
<body>
  <h1>CS142</h1>
  <div>Div #1</div>
  <div class="div2">Div #2</div>
  <div class="div3">Div #3</div>
</body>
</html>
```

(a)

CS142

Div #1

Div #3

(b)

CS142

Div #1

Div #3

(c)

CS142

Div #1

Div #3

(d)

CS142

Div #1

Div #3

(e)

CS142

Div #1

Div #3

1B (5 points)

You are building a web application that allows students at a university to vote for candidates for their student council. For each of the following pieces of logic that your web application needs, write which part of the MVC pattern would be responsible for handling it: Model, View, or Controller. Write a brief justification for your answer.

- (i) Store a student ID and password for each student so that they can log in to submit their vote:

- (ii) Handle a student submitting their vote and send the vote information to the web server:

- (iii) Display the names, photographs, and candidate statements for each student council candidate for voters to peruse:

- (iv) Keep track of how many votes each candidate has: _____

- (v) Show the results of the election once all votes have been submitted:

Problem #2 (8 points)

2A (5 points)

Consider the following URL:

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

Given the following components of a URL we learned in lecture, please identify them in the URL provided by underlining (note: since the underlines may come close to each other, you may want to alternate between squiggly underlines and straight underlines for visual clarity) each component and writing the corresponding letter underneath. If you believe the component doesn't exist in this URL, you may ignore it.

Server's port number (A)

Scheme (B)

Hierarchical portion (C)

Fragment (D)

Hostname (E)

Query parameters (F)

For example, if you believed “:80/class/cs142” was the server's port number, part of your answer would look like:

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`
A

2B (3 points)

Assume that the page fetched by the URL in Problem 2A is properly formatted HTML that contains several hyperlinks. For each of the hyperlinks underline the part of the URL that will change when the link is clicked:

`index.html`

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

`/index.html`

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

`#foo`

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

Problem #2 (8 points)

2A (5 points)

Consider the following URL:

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

Given the following components of a URL we learned in lecture, please identify them in the URL provided by underlining (note: since the underlines may come close to each other, you may want to alternate between squiggly underlines and straight underlines for visual clarity) each component and writing the corresponding letter underneath. If you believe the component doesn't exist in this URL, you may ignore it.

Server's port number (A)

Scheme (B)

Hierarchical portion (C)

Fragment (D)

Hostname (E)

Query parameters (F)

For example, if you believed “:80/class/cs142” was the server's port number, part of your answer would look like:

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`
A

2B (3 points)

Assume that the page fetched by the URL in Problem 2A is properly formatted HTML that contains several hyperlinks. For each of the hyperlinks underline the part of the URL that will change when the link is clicked:

`index.html`

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

`/index.html`

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

`#foo`

`http://web.stanford.edu:80/class/cs142/lectures/URLs.html?professor=Rosenblum`

Problem #3 (18 points)

3A (12 points)

If you run the following program:

```
var v1 = 0;

function f(v2) {
  var v3 = 0;
  return function (v4) {
    v1++; v2++; v3++; v4++;
    console.log(v1,v2,v3,v4);
  }
}

var f1 = f(1);
var f2 = f(5);
var f3 = f1;

f1(10); f1(12); f2(13); f3(14);
```

It will print out 16 numbers. In the table below write the 16 numbers that are generated by the console.log statement.

console.log from call to:	v1	v2	v3	v4
f1(10)				
f1(12)				
f2(13)				
f3(14)				

3B (3 points)

For each of the code fragments below state what the `console.log` message will print. If the execution aborts due to an error, describe the error.

(a)

```
var a = 3;  
console.log(a);
```

(b)

```
console.log(a);  
var a = 3;
```

(c)

```
var a;  
console.log(a);  
a = 3;
```


3C (3 points)

For each console.log function call below, write down its output on the right.

```
function Person() {  
    this.classmates = ['Amy', 'Bob'];  
}
```

```
Person.prototype.name = "Lina";  
Person.prototype.age = 35;  
Person.prototype.friends = ['Alice', 'Ellen'];
```

```
var person1 = new Person();  
var person2 = new Person();  
person1.name = "Nina";
```

```
console.log(person1.name)           //Output:_____
```

```
console.log(person2.name)           //Output:_____
```

```
person1.friends.push("Carol");  
person1.classmates.push("Nick");
```

```
console.log(person1.friends);        //Output:_____
```

```
console.log(person2.friends);        //Output:_____
```

```
console.log(person1.classmates);     //Output:_____
```

```
console.log(person2.classmates);     //Output:_____
```

Problem #4 (12 points)

Consider the following HTML and JavaScript program:

```
<!doctype html>
<html>
  <head>
  </head>
  <body>
    <div id="container">
      <button id="button">Click me!</button>
    </div>
  </body>
  <script>
    'use strict';
    function printWindow(event) {
      console.log('Window says hello!');
    }

    function printTarget(event) {
      console.log('Target says hello!');
    }

    function printContainer(event) {
      console.log('Container says hello!');
    }

    var capture = true;
    var button = document.getElementById('button');
    var container = document.getElementById('container');

    window.addEventListener('click', printWindow);
    button.addEventListener('click', printTarget, capture);
    container.addEventListener('click',
printContainer,capture);

  </script>
</html>
```

The page loads and the user waits a few seconds before clicking the button that says "Click me!". What's the console output?

~~Problem #5 (15 points)~~

5A (6 points)

In Angular, the two-way binding is implemented through watch and digest mechanisms. Describe these mechanisms and how they bind JavaScript state into the HTML templates.

5B (9 points)

Consider the Angular view template fragment:

```
-----
<div ng-controller="FirstController">
  <h1>{{first.selectedGreeting}}</h1>
  <h1 ng-if="!first.selectedGreeting">Not a valid index</h1>
  <div ng-controller="SecondController">
    Pick a greeting index!
    <input type="number" ng-model="first.greetingIdx">
    <button ng-click="displayGreeting(first.greetingIdx)">
      Select Greeting
    </button>
    <ol>
      <li ng-repeat="greet in greetings">{{greet}}</li>
    </ol>
  </div>
</div>
```

Hello

Pick a greeting index! 1

Select Greeting

- 1. Hello
- 2. Hola

Hola

Pick a greeting index! 2

Select Greeting

- 1. Hello
- 2. Hola

Not a valid index

Pick a greeting index! 3

Select Greeting

- 1. Hello
- 2. Hola

Implement `FirstController` and `SecondController` in the given code skeletons below, such that the given view template will show the behavior in the screenshots.

```
cs142App.controller('FirstController', ['$scope', function($scope) {
```

```
}]));
```

```
cs142App.controller('SecondController', ['$scope', function($scope) {
```

```
}]));
```

~~Problem #6 (12 points)~~

6A (6 points)

When using Angular Material to implement a responsive web design, it is possible to write a template like:

```
<md-button show-lg ng-click="doSomething();">  
    Do Something  
</md-button>
```

that will only be shown on large screens. Describe the browser mechanisms (i.e. HTML/CSS features) and how they are used by Angular to implement this responsive design functionality. Your answer should cover what Angular does when it encounters a template like this.

6B (6 points)

What is the difference between unit testing and end-to-end testing as applied to web applications? Provide an example of each type in your comparison.

Problem #7 (12 points)

7A (6 points)

Some web applications studiously update the browser's notion of the current location as the user moves around in the application, even though the updates don't change the browser state and are frequently not referenced by any code or person. Describe the possible reasons why web applications might do this.

7B (6 points)

Imagine you have a set of web applications running in a browser. Some of the web applications are considered to be single-page applications, while others are not. Describe how you could differentiate between the two types by examining their behavior in the browser