Sample CS 142 Midterm Examination

Winter 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 email account for grading database key)

Problem	#1	#2	#3	#4	#5	#6	#7	#8	Total
Score									
Max	12	12	12	10	12	10	12	10	90

Problem #1 (12 points)

Using the following HTML document say what the DOM accessing JavaScript expressions listed below would return.

```
<html>
 <body id="body">
  <table id="robots" class="goodBots">
    <tbody>
      <tr id="names" class="myHeaderClass">
        <th id="dolores">Dolores</th>
        <th id="bernard">Bernard</th>
        <th id="maeve">Maeve</th>
      </tr>
      <tr id="age" class="myRowClass">
        <td id="150" class="agesClass">150</td>
        <td id="findMe" class="agesClass">
          <span id="sneakySpan">100</span>
        </td>
        <td id="60" class="agesClass">60</td>
      </tr>
      <tr id="gender" class="myRowClass">
        <td id="femaleOne">F</td>
        <td id="male">M</td>
        <td></td>
      </tr>
    </tbody>
   </table>
  </body>
</html>
```

a) `document.getElementById("findMe").className`

b) `document.getElementById("findMe").parentNode.className`

c) `document.getElementById("findMe").parentNode.parentNode.parentNode.id`

Continued on next page...

…Continued from previous page

Hint: `nextElementSibling` and `previousElementSibling` are like `nextSibling` and `previousSibling` except they skip over DOM nodes representing the whitespace in the HTML.

```
d) document.getElementById("findMe").parentNode.
   nextElementSibling.id
```

```
e) document.getElementById("sneakySpan").parentNode.
   previousElementSibling.id
```

```
f) document.getElementsByTagName("td")[2].id
```

Problem #2 (12 points)

```
var object = {numProp: 1, stringProp: "foo", obj1Prop: {prop:
'foo'}};
object.__proto__ = { stringProp: "bar", obj1Prop: {prop: 'bar'},
   obj2Prop: {}};
print("1", object); print("1p", object.__proto__);

object.numProp = 2;
object.stringProp = "foo2";
object.obj1Prop.prop = "prop2";
object.obj2Prop.prop = "prop2"

print("2", object); print("2p", object.__proto__);

function print(tag, obj) {
  console.log(tag, "numProp", obj.numProp,
      "stringProp", obj.stringProp, "obj1Prop", obj.obj1Prop,
       "obj2Prop", obj.obj2Prop);
}
```

When the above code is executed it prints four lines to the console log. Each line contains four properties from either the object or its prototype. Fill in what this code would output in the form below. Assume console.log prints the entire object (e.g. `console.log({prop: 'foo'})` prints `'{prop: "foo"}'`.

Hint: The property __proto__ returns the prototype of an object.

1 numProp ________ stringProp ________ obj1Prop ________________ obj2Prop ________________

1p numProp _______ stringProp ________ obj1Prop ________________ obj2Prop ________________

2 numProp ________ stringProp ________ obj1Prop ________________ obj2Prop ________________

2p numProp _______ stringProp ________ obj1Prop ________________ obj2Prop ________________

Problem 2 continued…

```
var globalVar = 1;
function foo(argVar) {
  var localVar = 2;

  return function (arg) {
            argVar += arg;
            localVar += arg;
            globalVar += arg;
            console.log('F', argVar, localVar, globalVar);
        };
}

var func1 = foo(1);
var func10 = foo(10);
func1(1);
func10(2);
console.log('G', globalVar);
```

Show what the above code will output to the console log when executed.

Problem #3 (12 points)

Given the URL `http://web.stanford.edu/class/cs142/exams.html?v=win1617#midterm`

(A) Please fill in the blanks:

URL Part	**Example URL Section**
________________	`?v=win1617`
Host name	________________
________________	`http:`
Hierarchical portion	________________
________________	`#midterm`

(B) **Without** using a full URL, complete the HTML snippets below, making sure to write the **shortest possible** link to get to the URL above if:

The browser is currently at `http://web.stanford.edu/class`

```
<______________________________________________________________>
      Link A
</__________>
```

The browser is currently at `http://web.stanford.edu/schedule`

```
<______________________________________________________________>
      Link B
</__________>
```

(C) List **two** additional uses for a URL, beyond loading a page in a web browser.

(D) What is **referential integrity**? Does the modern web have it? Why or why not?

Problem #4 (10 points)

(a) Angular's Directive abstraction is intended to aid in building reusable components. Like most things accessed during the Angular compilation phase using a Directive will generate a new Angular Scope and and that Scope will be a child of where the directive is invoked. Angular also supports defining Directives that although they get a new scope it does not act as a child scope. The directive is given what is called an *isolated scope.* The Directive definition specifies the content. To support reusable components Angular strongly recommends that isolated scopes be used for all Directives. Briefly explain why Angular recommends this.

(b) Angular marketing material claims it supports "two-way binding". Briefly state what are the two ways including a description of the source and the destination of the "way".

Problem #5 (12 points)

Given the following simple HTML document that contains the word "Div" and some JavaScript that registers event handlers on it.

```
<html>
  <body id="body">
    <div id="div">Div</div>
 </body>
</html>

document.getElementById("body").addEventListener("click",
 function (e) {console.log('Body-true', e.target===e.currentTarget)},
 true);
document.getElementById("body").addEventListener("click",
 function (e) {console.log('Body-false',
e.target===e.currentTarget)},
 false);
document.getElementById("div").addEventListener("click",
  function (e) { console.log('Div-true',
e.target===e.currentTarget)},
  true);
document.getElementById("div").addEventListener("click",
  function (e) {
console.log('Div-false',e.target===e.currentTarget)},
  false);
```

(Hint: The documentation for `addEventListener` names its last parameter `useCapture`.)

(a) Describe what would be printed to the console.log if you clicked on the word "Div".

(b) Describe what would be printed to the console log if you clicked on the page but not on or near the word "Div".

Problem #6 (10 points)

(a) A browser-based URL routing package such as Angular's ngRoute wants to assign different URLs to different views so that browser navigation (e.g. forward/back buttons, history) can be used to navigate among the views. A browser normally restarts a fresh JavaScript Virtual Machine when navigating to a different page. Explain how a browser-based URL routing package can have different views without this JavaScript restart.

(b) Although a single page application can use browser navigation controls, the browser refresh button (e.g. ⟳) cannot be disabled so a good single page application will have the refresh restore the application to the place at which the refresh occurred. Describe how this functionality can be supported even in the face of the browser tossing all the JavaScript state.

Problem #7 (12 points)

For each of the following web application functionality items, state which component (model, view, or controller) you would expect to expand in size the most when you add the functionality item to a web page. Provide a brief justification of your answer and make sure your answer demonstrates that you know what the functionality item is.

(a) Responsive design

(b) Internationalization (I18N) support

(c) Accessible Rich Internet Applications (ARIA) support

Problem #8 (10 points)

```
<html>
<body>
  <div id="div1">Div1</div>
  <div id="div2" class="class1">Div2</div>
  <div>Div3
     <span id="span1" class="class1">span1</span>
     <span id="span2" class="class2">span2</span>
  </div>
  <div>Div4</div>
  <span>span3</span>
</body>
</html>
```

For each of the following CSS rule(s), write the words (e.g. “Div2”) that will appear red or green in the browser and specify the color. If a word does not have a red or green color you should not write it down. Rules from one part do not apply to the next part. Briefly justify your answers for each part by stating what the CSS rules are doing.

```
(a)   div { color: red}
```

```
(b)   div.class1 { color: red}
```

```
(c)   .class1 { color: red}
```

Problem continued on next page….

….Continued from previous page

```
(d)   #span2 {color: red}
```

```
(e)   div { color: red}
      div.class1 {color: green}
```