CS142 Fall 2017 Midterm Exam Rubric

Problem #1 (13 points)

1A (8 points) - Hana

Answer: The correct answer is (b). To answer this question correctly, the student must understand the following concepts:

- "Visibility: hidden" still allocates space for the hidden element, though the element is not visible. This contrasts with "display: none" (used for option (d), which is otherwise correct).
- While the "border" property is not inherited from parent to child element, div #3 still has a border because it is not a child of div; it **is** a div, merely with its own class. Some students may mistakenly choose (e) because they remember "border" to be one of the properties that is not inheritable.
- In CSS, specific rules override general. The specific "font-size: 20" for div #3 overrides the general "font-size: 10" for the body. Likewise, the specific "border: 1px solid black" for the divs overrides the general "border: 2px solid blue" for the body.

Rubric: Full credit for (b), no credit for incorrect answers

1B (5 points) - Hana

Answer: The correct answer should be: Model, Controller, View, Model, View.
Rubric: -1 points for each incorrect answer, partial credit for compelling explanations. -0.5 points for each correct answer with no justification/wrong justification.

Problem #2 (8 points)

2A (5 points) - Jason

Answer:

http:	//web.stanford.edu	:	80	/class/cs142/lectures/URLs.html	?professor=Rosenblum
B	E		A	C	F

Server's port number (A)
Scheme (B)

Hierarchical portion (C)
Fragment (D)
Hostname (E)
Query parameters (F)

Rubric: -1 mark for each part. For hostname, “//” can be included or not.

2B (3 points) - Jason

Answer:
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:

<a href="index.html">index.html</a>

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

<a href="/index.html">/index.html</a>

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

<a href="#foo">#foo</a>

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

Rubric: 1 marks for each example

Problem #3 (18 points)

3A (12 points) - Nina

Answer:

v1	**v2**	**v3**	**v4**
1	2	1	11
2	3	2	13
3	6	1	14
4	4	3	15

Rubric:
0.75 point per answer
[-3] almost correct, but off-by-one for all answers

3B (3 points) - Nina

Answer:

a) 3
b) undefined
c) undefined

Rubric: 1 point per answer

3C (3 points) - Nina

Answer:

"Nina"
"Lina"
["Alice", "Ellen", "Carol"]
["Alice", "Ellen", "Carol"]
["Amy", "Bob", "Nick"]
["Amy", "Bob"]

Rubric: 0.5 point per answer. No point deducted for incorrect array format.

Problem #4 (12 points) - Ellen

Answer:

Container says hello!
Target says hello!
Window says hello!

Rubric:

[-4] window first
[-4] missing any (-4 per missing)
[-4] flipped target and container
[-6] combined capture and bubble phase
[-6] for claiming anything prints on page load

Problem #5 (15 points)

5A (6 Points) - Marcella

Answer:

Angular add a watch for every variable or function in template expressions. During the digest processing all watched expressions are compared to their previously known value and if different the template is reprocessed and the DOM update.
(From lecture slide "Introduction to AngularJs" pg. 16)

Rubric:
(6/6) Clean explanation about **watch** and **digest**.
(3/3) Wrong/flipped explanation, but still give a description
[-3] Inaccurate explanation
[-2] for missing watch / digest.
[-1] no explanation on how the HTML is updated with the JS state

5B (9 Points) - Marcella

Answer:

```
cs142App.controller('FirstController', ['$scope', function($scope) {
        $scope.first = {};// Need to be initialized before setting $scope.first.<prop>
        $scope.first.greetingIdx = 1; // can be any value
}]);
cs142App.controller('SecondController', ['$scope', function($scope) {
        $scope.greetings = ['Hello', 'Hola'];
        $scope.first.selectedGreeting =  $scope.greetings[$scope.first.greetingIdx -
        1]; // can also initialize in FirstController, and can be any value
        $scope.displayGreeting = function(strIdx) {
                var idx = parseInt(strIdx); // strIdx also works
                if (idx - 1 < $scope.greetings.length) {
                        $scope.first.selectedGreeting = $scope.greetings[idx - 1];
                } else {
                        $scope.first.selectedGreeting = ""
                }
                // This is also acceptable,  because $scope.greetings[idx - 1] ==
                // undefined
                // $scope.first.selectedGreeting = $scope.greetings[idx - 1];
        }
}]);
```

Rubric:
FirstController (3 Points)

[-2] Not initializing $scope.first, directly assigning $scope.first.greetingIdx, etc.
[-3] FirstController not implemented

SecondController (6 Points)
[-1] indexing off by 1
[-1] test for invalid number is wrong (e.g. idx <=1 || idx >=3)
[-1] equality (== vs ===)
[-1] array declared as object/dictionary
[-1] Wrong array values ('1. Hello', '2.Hola'). The <li> already provides the index
[-1] Not a valid parameter declaration for displayGreeting. (i.e.displayGreeting($first.greetingIdx)
[-1] unnecessary $scope.$digest
[-1] incorrectly referencing $scope.first
[-2] selectedGreeting is a boolean
[-2] Not handling invalid index
[-2] Not defining the array ['Hello', 'Hola'] for the list HTML
[-6] SecondController not implemented

Problem #6 (12 points)

6A (6 Points) - Jeff

Rubric and Answer:
Angular uses media queries/CSS breakpoints (get 6pts)
Doesn't mention media queries, but describes them (-1 points)
Description mentions CSS breakpoints but not media queries or is brief/incomplete (get 4pts)
Describes mechanism but does not mention @media queries or CSS breakpoints (get 2pts)

6B (6 Points) - Jeff

Example Answer:
Unit testing is testing small modular parts of an application by themselves to make sure they operate as expected. e.g. testing the inputs/outputs of a Javascript function. This should be restricted to one component.
End to end testing tests the whole application as an integrated system, e.g. testing that a photo can be uploaded and viewed. It involves the frontend and backend usually. Or multiple components of an app working together.

Rubric:
Unit testing: Specific components (worth 2 pts)

- example (worth 1pts)

E2E: Communicate with web app (full stack) (worth 2pts)

- example (worth 1pts)

Problem #7 (12 points)

7A (6 Points) - Mendel

The browser's notion of the current location, shown in the URL bar in some browsers, is accessed when the user bookmarks the current page, copies the URL for saving or sharing, or pushes the refresh button on the browser. A web application might update the location continuously so to be ready if the user does any of the above events. The idea is the captured URL would do something useful in above the cases.

Having a web app that supports "deep-linking" is required for replying the URL to do something useful but updating the URL isn't necessary for deep-linking to work. Consider the case of having a special share button that generates a special URL for doing "deep-linking".

7B (6 Points) - Mendel

From observations of a web application running in a browser, a single page application would fetch a page to start, start a JavaScript environment, and then not tear the environment down until the web application finishes. The lack of environment teardown would be a good predictor of a single page application.