CS142 Winter 2017 Midterm Grading Solutions and Rubric

See http://web.stanford.edu/class/cs142/info.html#regrades for instructions on correcting grading mistakes.

Problem 1 (Whitney)

a) “agesClass”
b) “myRowClass”
c) “robots”
d) “gender”
e) “150”
f) “60”

Problem 2 (Don)

Part (i) solution:
1 numProp **1** stringProp **foo** obj1Prop **{ prop: 'foo' }** obj2Prop **{}**
1p numProp **undefined** stringProp **bar** obj1Prop **{ prop: 'bar' }** obj2Prop **{}**
2 numProp **2** stringProp **foo2** obj1Prop **{ prop: 'prop2' }** obj2Prop **{ prop: 'prop2' }**
2p numProp **undefined** stringProp **bar** obj1Prop **{ prop: 'bar' }** obj2Prop **{ prop: 'prop2' }**

Part (ii) solution:
F 2 3 2
F 12 4 4
G 4

For part (i) (8 points, -8 max):
-0.5 pt for each incorrect blank

Common mistakes:
People wrote in {} for 2p, obj2Prop

For part (ii) (4 points, -4 max):
-1 pt per incorrect line
-1 additional pt if incorrect number in any of the two F lines
-1 pt for extra console.log lines

Common mistakes:
People wrote in “12 5 4“ for the 2nd F line

Problem 3 (Jeff)

a) (**4 points) (-1 each error)**

Query parameters	?v=win1617
Host name	//web.stanford.edu
Scheme (also accepted: protocol)	http:
Hierarchical portion	/class/cs142/exams.html
Fragment (also accepted: anchor)	#midterm

b) **(4 points)**
   i) Link A
   ii) Link B

c) **(2 points)**
   i) (1 point) Required: another scheme besides http (e.g. mailto, file, ftp, ssh), and/or another port besides 80.
   ii) (1 point max) Other answers:
      1) Locating server API to fetch data
      2) Locating page resources (images, stylesheets)

d) **(2 points)**
   i) A concept where every web page has a unique identifier associated with it and every valid identifier leads to a valid web page. This is not the case on the modern web (designed by physicists), valid URLs may lead to invalid webpages (404) or the resources at an endpoint may change.

Common Mistakes:
Using the word “spreadsheets” instead of stylesheets. Incorrect understanding of relative URLs. Not including the // in the host name. Only mentioning uses of URLs related to fetching a page in a web browser (such as bookmarking a website URL, or sharing a website URL with another person). Incorrect understanding of relational integrity. Only mentioning relational integrity as it relates to databases, not as it relates to referencing web pages on the world wide web.

Problem 4 (Kunmi)

a) Using isolated scopes forces directives to be implemented as a self-contained modular unit. Since they do not depend on any data outside the isolated scope, they can be

easily plugged in anywhere in your HTML without fear of having different behaviors in different contexts.

Full points (5/5) - To get full credit, what is above needs to be expressed clearly

Partial Points (3/5) - Many people talked about the interaction of the scope of the directive with its parent scope and how isolating the scope prevents the directive's child scope from tampering with the parent scope. Although this is true, it doesn't directly address the reusability point in the question.

Other deductions - Extra points were taken off when students expressed the right idea but included statements that were wrong.

Common mistakes - Some thought that directives will share the isolated scope.

b) Two-way binding refers to the automatic synchronization between the model and the view. Changes in the application data in the model(source) are reflected in the view(destination) which is displayed to the user. Updates made in the view(source) are also automatically reflected in the model.

Full points (5/5) - [We were quite lenient with this question] Full points were awarded as long as the student demonstrated a general understanding of "two-way binding" and provided an explanation of what the model and view are. We accepted things like template/DOM in place of view and controller in place of model (although these are not technically not the right terms to use). Descriptions of the way the binding occurred instead of the source and destination were also accepted.

Partial Points (4/5) - Displays understanding but no description of source and destination or way binding occurs.

Other deductions - Extra points were taken off in when students expressed the right idea but included statements that were wrong.

Common mistakes - Binding between template and DOM. Binding between child scope and parent scope.

Problem 5 (Shannon)

a) Answer (8/8):
Body-true false

Div-true true
Div-false true
Body-false false

Explanation:
The capture phase runs before the bubble phase, so all the "*-true" handlers must print first. Within the capture phase, the <body> click handler is triggered first. Within the bubble phase, the click handler is triggered first. e.target is always the div element.

Common mistakes:
Extra print statements from the <html> element, leaving out the "body-*" or bubble phase event handlers, placing all the body event handlers before the div event handlers.

b) **Answer (4/4):**
Body-true true
Body-false true

Explanation:
Only the <body> click handlers are called. The capture phase runs first, and e.target is always the body element.

Common mistakes:
Extra print statements from the or <html> elements, missing the bubble phase event handler.

Grading:
2 pts for each line (1 pt each for the correct callback and the correct target expression)
-1 pt for each incorrect, missing, or additional part
-0.5 pts for writing "div" or "body" instead of "true" or "false"

Problem 6 (Kevin)

c) **Full points (5/5)** - the answer mentioned that routing packages either
   I. Intercept the links so that the browser doesn't handle them
   II. they take advantage of parts of the url that don't perform full page reloads (i.e. fragments).

Partial Points (2/5 or 3/5) depending on how far off the answer was. A lot of students described how to use ngRoute from AngularJS in an application but didn't go into how AngularJS is able to do that (as mentioned above in the **Full Points** answer). Other common answers included mentioning deep linking, accessing window.location, etc. All

of these got partial points.

d) **Full points (5/5)** - deep linking or saving the state of the application in the url were mentioned.

   Partial Points (3/5) none of the above were mentioned. You received partial points if you mentioned:

   I. Storing the state of the application on the server
   II. Browser caching
   III. Cookies
   IV. Saving the state in the scope

Problem 7 (Mendel)

(a)

In **Responsive Design** we build the web application to have views that handle different screen sizes and resolutions. The examples in class we saw included using CSS to trigger different layouts and the concept of CSS breakpoints. These are part of the view component so we would expect to have larger **view components**.

(b)

Internationalization (I18N) support requires displaying different information based on the viewer's choice of language and cultural conventions. When only one language needs to be support it is easiest to hard code text in the HTML template. With I18N support this text becomes input to the HTML template so acts like model data. One would expect more **model components**.

(c)

Accessible Rich Internet Applications (ARIA) support refers to an approach to making web app accessible to people with disabilities. One part of approach is to include text descriptions of images, links, and other document components that someone with vision problems might have trouble using. This would result in more information added to the **view component**.

Each part was assigned four points. Partial credit was given as follows:

A. If the answer's justification provided no evidence of knowledge of what the asked about functional item is, no credit was given.
B. Between zero and two points were taken off for answers that contained an indication of knowledge of the functional item but choose a different component. The amount of the deduction depended on how plausible and persuasive the justification text was.

Problem 8 (Alex)

a) **Red**: Div1, Div2, Div3, Div4, span1, span2
Explanation: Text in all divs (including children) turn red

b) **Red**: Div2
Explanation: Only a div with a class of “class1” has the style applied

c) **Red**: Div2, span1
Explanation: ANY element with class of “class1” has style applied

d) **Red**: span2
Explanation: Only element with id = “span2” has style applied

e) **Red**: Div1, Div3, Div4, span1, span2
Green: Div2
Explanation: Text in all divs (including children) turn red. Text in div with class of “class1” will become green due to CSS priority rules