CS142 Spring 2016 Midterm Exam Solutions

Question 1 - Graded by Brandon

10 points total. 1 point per log.

log 1: undefined
log 2: 3
log 3: 3
log 4: 3
log 5: 4
log 6: 4
log 7: 5
log 8: 80
log 9: 4
log 10: undefined

For more details about the solution try typing the JavaScript into a file and running in under Node.js.

Question 2 - Graded by Don

Answer:
(a)
log 1:
0, 2, 3

log 2:
0, 6, 7

log 3:
5, 6, 7

log 4:
9, 8, 7

(b)
To receive full points, need to give a valid non-trivial difference between adding a method via the prototype pattern and defining a method inside of an object's constructor:
e.g., one of:
1) If a method is added via the prototype pattern, that method's definition is shared across all instances. If the method is instead assigned to the constructed object in the object's constructor, then the method is defined separately for every instance, leading to additional memory consumption per created instance
2) The body of a method defined via the prototype pattern does not have access to the local variables that are declared in an object's constructor function
3) Methods defined in an object's constructor function are not part of the prototype chain

Rubric:
(a)
-0.5 for each incorrectly filled slot

(b)
-5 for incorrect answer
-2 for incomplete/unsatisfactory answer and/or incorrect statement(s)

Problem 3 - Graded by Kevin

Part A:
Answer: second and fifth box

6/6:
- circled the correct ones and no extra

4/6:
- Circled correct ones and one extra incorrect one
- Circled one correct one

3/6:
- circled correct ones and two extra

2/6:
- circled correct ones and three extra

0/6:
- none correct

Part B
First Question
3/3:
- Correct Answer(Capture Phase) with Correct Explanation (capture phase is always prioritized over bubble no matter what)

2/3:
- Correct Answer(Capture Phase) with reasonable explanation but did not explain what is prioritized

1/3:
- Correct Answer (Capture Phase) with no explanation

0/3:
- Incorrect Answer

Second Question
3/3:
- Correct Answer(Capture Phase) with Correct Explanation (capture phase is always prioritized over bubble no matter what and clicking on child element does not change that)

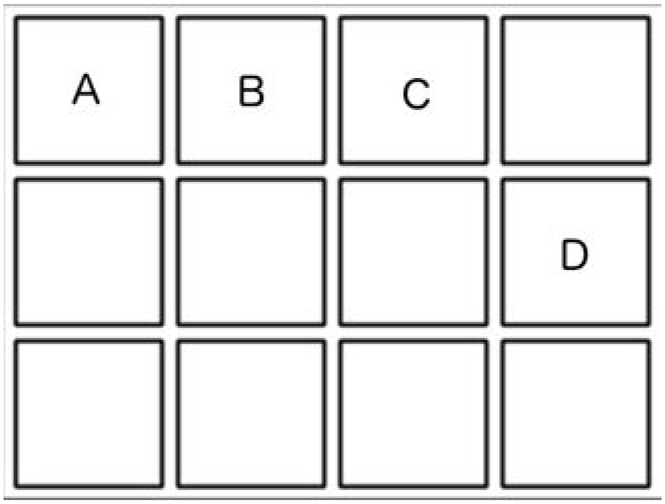
2/3:
- Correct Answer(Capture Phase) with reasonable explanation but did not explain what is prioritized

1/3:
- Correct Answer (Capture Phase) with no explanation

0/3:
- Incorrect Answer

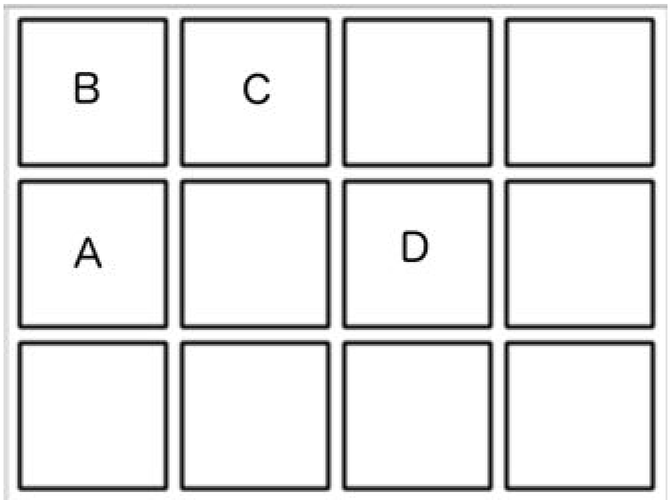
Problem 4 - Graded by Jed

A)



A	B	C	
			D

B)



B	C		
A		D	

Rubric:
-2 for a misplaced element
-3 for multiple misplaced elements caused by one mistake
(i.e. having B, C next to A, D in bottom right corner for part B).
-5 maximum for one grid if the student has any element correct (i.e. having B in the top left corner for part B, but A, C, D in the wrong place).

Problem 5 - Grade by Jennifer

Part A (4 points):
scope and HTML or model and view
Changes to the view updates the model and vice versa
Part B (2 points):
watch and digest (refer to Lecture AngularJS Slide 16)
Angular adds 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

Part C (3 points):
Without ng-app, the application will not recognize that it is an Angular application and will lose its scope. All connections with Angular controller and model will be lost and only the raw HTML will be displayed. For instance {{name}} will be displayed literally as such instead of filling in with the value of the variable name.
(needed to include both a description of what ng-app is for and what is displayed)

Problem 6 - Grade by Raymond

A (3 points)

Modern frameworks use CSS breakpoints to specify rules for *ranges* of screen sizes. These ranges essentially bucket devices into categories. This ensures that a CSS stylesheet dynamically applies across all ranges of screen sizes. For example, Bootstrap uses (max-width: 768px) for smaller devices such as phones.

The key phrases in this question was “across all device sizes” and “media queries.” Full credit solutions discussed CSS breakpoints and grouping screen sizes into ranges for different types of devices in the context of media queries.

Some students described using relative sizes such as em, rem, and % instead of absolute sizes such as px. While using relative sizes are part of responsive design, relative sizes are more related to layout and the question specifically asked about media queries. Some students also discussed using a grid layout and employing a mobile first design. Both of these are related to responsive design but are not specific enough to media queries. These answers were given partial credit.

No credit was given for answers that simply regurgitated the definition of a media query given in the question.

B (3 points)

CSS applies all the rules from both declaration blocks to the particular DOM element. If two rules conflict (i.e. both declaration blocks have rules for the same style), it applies the rule that belongs to the more specific selector.

Full credit was given to answers that discussed how CSS applies the more specific selector. A common mistake was an answer that described how CSS applies the selector that appears later in the document. This is true if and only if the selectors have the same level of specificity.

C (3 points)

Block elements start on a new line and take up the entire horizontal width available by default. They can also have a specified height and width.* Inline elements take up as much space as their content, or in other words they only take up as much space as necessary. This allows them to be placed along the same horizontal line, wrapping around if necessary.

*did not need to specify this detail for full credit

There were many common answers for this question that were correct but not complete. Partial credit was given for these answers:

- Block elements are displayed vertically while inline elements are displayed horizontally.
This answer shows a surface level understanding but does not address how the elements occupy space.

- Block elements can have their height/width specified.
This answer does not address the default behavior of block elements, which is that they occupy the entire horizontal width available. It is important to understand the default behavior.

- Block elements behave like divs while inline elements behave like spans.
This answer does not explain the behavior of the elements, it just provides examples.

- Block elements start on their own line / create line breaks before and after the element.
- Correct, but does not address the behavior that block elements occupy the entire horizontal width available.

No credit was given for answers that relied solely on the words “block” or “inline” / “in a line” for their explanation, e.g. “display block displays the elements like a block while display: inline displays the elements in a line.” No credit was given for answers that described the positioning of the element rather than the display of the element (how the element occupies space).

D (3 points)

The element is positioned relative to its static/default/normal position in the document flow.

This question was all or nothing. The most common mistake was confusing position: relative with position: absolute. Other common mistakes were that the element is positioned relative to other elements with the same class name, or its nearest sibling element.

Problem 7 - Graded by Santiago

(A) (2 points)

must be represented as %xx, where xx is the hexadecimal value of the character:

(B) (6 points)
(1 point for each part, including description)
file://host.cs142.com:50/animal/bird/penguin.html?user=Mendel#h3

file:// -> scheme, identifies protocol used to fetch the content.

host.cs142.com -> Host name, name of a machine to connect to

:50 -> Server's port number, allows multiple servers to run on the same machine

/animal/bird/penguin.html -> path or Hierarchical portion, used by server to find content

user=Mendel -> Query parameters, provides additional parameters

#h3 -> Fragment, Have browser scroll page to fragment

Problem 8 - Graded by Mendel

(A-1) (3 points) Deep-linking refers to a web applications that provides the ability for the user to use URLs to directly access context in inside of the application without having to navigate inside the application. This capability permits the user to bookmark and share URLs like they would do for websites.

Many people answered that it was storing the apps state in the URL. That is one implementation technique for deep-linking but not the only way of doing it.

(A-2) (3 points) To front-end developers response means web application adjusted itself based on the device characteristics.

(b) (4 points) The basic approach involves some kind of indirection so language or cultural specific things are parameters to the template rather than hard coded in the template.

Problem 9 - Graded by Shannon

Answer:
A) <b> and <i> are tags that define styling (bold/italic), and it's good practice to enforce separation of content (HTML) and style (CSS). These can be replaced by <strong> and <em> tags, which are semantic.

B)
<script> (JavaScript, including the Angular library, Angular controllers, etc)
<link> (links to CSS sheets, favicon images, etc)
<title>
<meta>
<style>

Rubric:
A) 3 pts
+3 for stating that <b> and <i> are style elements and we want to separate content from style.
+1 for each of the following (if separation of content/style not mentioned) :

- It's more efficient/flexible to change styling in CSS
- They make the HTML harder to read

B) 3 pts
+ 1 for each *distinct* element mentioned