

CS 142 Midterm Examination

Spring Quarter 2022

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 (10 points)

HTML is a markup language designed to support transferring documents around the world wide web using the Internet in its youth. Some Internet connections in those early days operated at speeds in the range of 30 to 40 bytes per second (i.e. 300 baud modem). A markup language works by extending the document's content with annotating tags so the size of an HTML document is the size of the content plus the size of the tags. To reduce the size increase, HTML made common tags (e.g. <p> and <a>) as small as possible.

- A. Given the original document's text content size and the size of all the HTML tags, we can compute an estimate of the total size of an HTML encoding of the document. Given your knowledge of HTML, how far off might this estimate be for the HTML encoding of an arbitrary document. Be sure to explain your answer.

- B. With the introduction of CSS, CSS style sheet bytes needed to be transferred over the Internet along with the HTML document bytes in order for the document to be displayed correctly. Like HTML tags, CSS stylesheets are relatively space-efficient. Would a highly styled HTML document (e.g. a document that overrides the browser's default styling) have a larger total transfer size including the CSS stylesheets when compared to doing the styling without CSS style sheets? Explain your answer.

Problem #2 (5 points)

Although the original HTML design was based on previous markup languages that were used to describe the formatting of documents, the HTML standard has moved away from using HTML for that role. Describe what the role of HTML is today if it isn't specifying the document's appearance.

Problem #3 (5 points)

Using CSS styling, a DOM element and its children can be marked invisible using two different CSS properties:

- Setting `visibility: hidden;`
- Setting `display: none;`

Both of these operations are reversible (e.g `visibility: visible;` or `display: block;`) so by periodically updating the DOM element we can cause part of an HTML to disappear and then reappear giving the user the illusion that the element is blinking. Which of the two approaches do you think would be more common to making something blink in a web application? Justify your answer.

Problem #4 (8 points)

Given the following HTML document body:

```
<body class="b">
  <div class="d1">
    Div1
    <div class="d2">
      Div2
      <div class="d3">
        Div3
        <p class="p">Paragraph</p>
      </div>
    </div>
  </div>
</body>
```

A. If you had a CSS rule that looked like:

```
.p { position: absolute; top 2px; left 2px;}
```

State the class of the element that the paragraph element is positioned relative to.

Explain your answer.

B. Assume you wanted the paragraph to be positioned relative to the div with class="d2". Show a CSS rule that would make that happen. Include an explanation of why it works.

Problem #5 (8 points)

Part A.

In project 4, we end up starting the web application by going to the location <http://localhost:3000/p5.html> but used React to generate the view. The view that React generated has hyperlinks that look like `States`. Identify the following components of the URL the browser goes to when that hyperlink is clicked.

Scheme:

Host name:

Port number:

Hierarchical portion:

Query parameters:

Fragment:

Part B.

Describe the processing done by the browser when it is given the above URL. Note this is asking what the *browser* does, not what the JavaScript environment does.

Problem #6 (8 points)

- A. When declaring a variable in JavaScript, the type of the variable is not specified. This is what the programming language experts call an *untyped* language. Although it is an untyped language, all JavaScript variables do have a type. How can the programmer know the type of a variable since it is not specified as part of the declaration?
- B. JavaScript has some weirdnesses in it. If you take the number 9007199254740991 and add 1 to it in JavaScript, you get 9007199254740992. But if you add 2 to it, you will also get 9007199254740992. This is unlike most other programming languages and calculators where you get 9007199254740993. Explain why JavaScript can't add 2 correctly here.

Problem #7 (8 points)

Part A.

Some React components that have an event handler method named `clickHandler` contain a line in the component's constructor that looks like:

```
this.clickHandler = this.clickHandler.bind(this);
```

This statement is being executed as part of an object allocation in which multiple objects are accessed or created. The objects include the newly allocated **component instance object** and the **component class prototype object**. List the sequence of object accesses done to these two objects when the above JavaScript statement is executed. Consider object reads that occur but return undefined. It could take less than 5 steps so leave any extra numbers blank.

- 1.
- 2.
- 3.
- 4.
- 5.

You can use the following words in bold in your steps:

- **Read instance** - read the 'clickHandler' property from the component instance object.
- **Read prototype** - read the 'clickHandler' property from the component class prototype object.
- **Write prototype** - write or create the 'clickHandler' property in the component class prototype object.
- **Write instance** - write or create the 'clickHandler' property in the component instance object.

Part B.

Describe the problem that led JavaScript programmers to start method functions by creating an alias name for the `this` variable such as: `let self = this;`

Problem #8 (6 points)

Considering using the JavaScript class keyword like:

```
class Foo {  
  methodA() { console.log("methodA"); }  
  static methodB() { console.log("methodB"); }  
};  
let f = new Foo();
```

When executing the above code the JavaScript runtime ends up creating three functions (Foo, methodA, and methodB). The function Foo is directly accessible in the current scope (i.e. `typeof Foo == 'function'`) but the methodA and methodB references are stored in different objects associated with the class. Describe what object each of the methods is stored in and show a JavaScript expression that would call the function.

A. methodA:

B. methodB:

Problem #9 (6 points)

Assume you are given a HTML document with the following body:

```
<body class="b">
  <div class="d1">
    Div1
    <div class="d2">
      Div2
      <div class="d3">
        Div3
        <p class="p">Paragraph</p>
      </div>
    </div>
  </div>
</body>
```

- A. How deep would the DOM tree be for the above HTML document? Justify your answer by listing the class names starting at the root and going to class p.

- B. Describe a change you could make to the HTML document that would extend the depth of the DOM tree by two levels?

Problem #10 (6 points)

- A. Can a capture phases event handler prevent a bubble phases event handler on the same event from running when the event is raised? If so, explain how. If not, explain why not.
- B. Can a bubble phases event handler prevent a capture phases event handler on the same event from running when the event is raised? If so, explain how. If not, explain why not.

Problem #11 (5 points)

One of the challenges of web applications is they need to work in the browser that is provided by the user. Even though web standards for HTML and JavaScript have moved forward rapidly, web applications can not assume the user's browser has been updated to track these standards. In fact, it is not uncommon for browsers to be years behind the standards.

In the first generation of web applications using static HTML documents, browser compatibility hampered the ability to use new features of HTML. React.js has eagerly embraced using new JavaScript features. Explain why React.js can get away with doing this.

Problem #12 (5 points)

In React when can register event handlers on 'click' events and use that mechanism to support switching views. When the user clicks on something the event handler can cause the new view to be rendered. When using React to build single applications we don't use this approach. Describe the approach we use and what advantages it has over using event handlers for view switching as described above. List at least two advantages.

Problem #13 (5 points)

Explain how the CSS grid system (`display: grid;`) helps in responsive web design.

Problem #14 (5 points)

HTML templates have long been used to specify views into web applications. Developers like that the templates make it easy to see the HTML structure of the view and where and how the dynamic data fits in. Explain how support for internationalization interferes with some of these benefits of templates.