1.1 Describe in your own words how the web works! In as much detail as you can, describe **all** the sequences of events that take place from the time a user presses Enter on the keyboard after typing in www.rpi.edu into the address bar to when the webpage is finished rendering in the browser. Specifically, tell me in great detail the protocols in action. (10 points)

...

1.2 What is the difference between a property and a method in JavaScript? (3 points)

The difference between a property and a method in Java Script is what they do for a JavaScript object. A property is an attribute that defines the characteristics of an object and how it behaves, such as the size and color of an element. A method is an action that can be performed on an object.

1.3 Explain how your browser chooses which CSS rule to apply to a tag in the case where there are multiple rules that could apply. (3 points)

There are three forms of CSS that can be applied to a tag: in-line CSS, in-file CSS, and CSS stored in a separate file. As your page gets loaded in your browser, the browser first applies the CSS from the external file (if the external file exists). Next, the in-file CSS, stored under the <style> tag, is applied, and it overrides any of the CSS from the external file if a common selector is mentioned. Finally, there is in-line CSS, which is called directly in a tag, that trumps all other CSS called throughout the file if a common selector is shared. So, the order of importance goes in-line CSS, in-file CSS, and CSS stored in a separate file, with in-line CSS being the most favored.

1.4 State **four** total advantages of "separation of concerns," for any permutations of that term we discussed in class. (4 points)

The concept of "separation of concerns" has many advantages. For example, if all of the code for a specific function of the web application is separated out, and one wants to make a change, much less code will have to be changed, compared to if the contents were not separated. Another advantage would be is if you want to swap a new implementation of your code, you could much easier, as you wouldn't have to fully understand the rest of the code to make the change, as well as it would be easier to locate the code that needs to be changed. Also, splitting up concerns helps avoid breakage in unrelated features by preventing you from having to change unrelated code. Finally, following the "separation of concerns" concept, the reusability of your code is greatly improved, as you can replace existing components of your architecture (i.e. a database) with much less hassle.

Extra Credit:

Professor Plotka graduated from RPI in 1987 with a degree in Computer Science.