Spring Semester 2013

251-0374-00

Web Engineering

09:00-11:00

13 August 2013

Answer all 4 questions.
The answers have to be in English.
Each question is worth a total of 20 points.
The points for each part of a question is given in brackets at the end of the part.

1. (a) Describe the basic structure of a WordPress theme in terms of the files that typically occur. (3)

   (b) How would you explain the concept of the WordPress Loop to a computer science colleague unfamiliar with WordPress? (3)

   (c) Show how you would create a custom loop in WordPress by using a `WP_Query` object to return the 3 most recent posts in the events. (2)

   (d) What is the difference between using the `query_posts` function for a custom loop instead of creating a `WP_Query` object? (2)

   (e) Explain the difference between `query_posts` and `get_posts`. (2)

   (f) Give an example to explain why you might need to reset the Loop data and how you would do this. (2)

   (g) What are the advantages of integrating custom post types into a theme instead of creating them using a plugin? (2)

   (h) Using an example, explain the steps involved in integrating a custom post type into a theme. Your answer should clearly indicate what files would need to be edited, what types of code changes or additions would need to be made to these files and what the purpose of these changes would be. It is not necessary to specify the details of the code. (4)

2. (a) Given the following HTML body and associated CSS stylesheet, sketch the layout of the web page. Your sketch does not need to be accurate to scale or show the colours of the elements, but it should clearly show the relative positions of the elements (including the text) and also of the background colouring. (4)

```
<body>
  <div id="container">
    <div id="outer">
      <div id="A"><p>element A</p></div>
      <div id="B"><p>element B</p></div>
      <div id="C"><p>element C</p></div>
    </div>
    <div id="D"><p>element D</p></div>
  </div>
</body>
```

```
#container { width:960px; margin:auto; background-color:#000000; }
#outer { position:relative; padding:80px; }
#A { float:right; width:350px; height: 300px;
     background-color: #ff0000; }
#B { float:left; width:450px; height: 150px;
     background-color: #00ffff; }
#C { float:left; width:200px; height: 150px;
     background-color:#ff00ff; }
#D { position:absolute; top:120px; width:300px; height:40px;
     background-color:#ffff00; }
```

   (b) Show how the layout would change if the final CSS rule was replaced with

```
#D { position:relative; top:120px; width:300px; height:40px;
     background-color:#ffff00; }
```

   Explain the changes to the layout and the reasons behind them. (3)

   (c) How would you change the CSS code, if it was decided that element D should be displayed instead as a footer bar positioned immediately below the other elements and extending the full width of the page? (1)

   (d) What CSS code changes would be made for a responsive design that scales the elements to cater for smaller viewports and changes to a linear vertical layout of the elements for mobile phones (landscape or portrait)? (3)

   (e) Explain what CSS pseudoclasses are, making it clear how they differ from other classes. (2)

   (f) Show how pseudoclass selectors could be used to generate a table where the rows alternate between light and dark grey. (2)

   (g) Explain the concepts of a sidebar and an HTML5 aside element, making clear what the differences are and how they might relate to each other. (2)

   (h) Developers should cater for the fact that HTML5 and CSS3 are evolving standards not yet supported by all browsers. Give an example to illustrate a specific problem and associated solution. (3)

3. (a) Many modern, server-side web application frameworks follow a design philosphy called "Convention over Configuration", especially with respect to model definitions. Give **one concrete example** of where this principle applies in either the Rails or django web framework. You may or may not use code to illustrate your example. (2)

   (b) Name **three** common features of server-side web frameworks and briefly describe their purpose and how they work. (6)

   (c) In the lecture, you have seen how web applications and services can provide parts of their web API and functionalities to web designers with no technical knowledge (non-programmers). Briefly describe how this was accomplished and give **two** examples of how this idea is being used in practice. (2)

   (d) i. In your own words, explain what OAuth is and what it is being used for. What is its main advantage? (3)

   ii. Sketch (i.e. draw) the full OAuth core authentication flow (version 1.0a) as it was presented in the lecture and list all the actions involved for a web application to access a user's protected resources. You should use the following roles/entities in the sketch: Consumer (i.e. the web application) and Service provider (e.g. Flickr).
   Please do not forget to correctly label the requests/responses exchanged between the entities and make sure the order of actions is obvious in your sketch. However, it is not necessary to list the individual parameters for each request/response. (7)

4. (a) Explain what a scroller unit is in WebML, giving an example of where it might be used. (3)

   (b) Explain what a transport link is in WebML, giving an example of where it might be used. (3)

   (c) Explain what content delivery networks are and how they operate. What role do they play in the development of web sites using client-side scripting? (4)

   (d) Explain the relationship between JSP and Java Servlet technologies. (2)

   (e) Explain what a JSP tag library is. (2)

   (f) Expain the differences between model-driven and interface-driven web engineering, outlining the relative advantages and disadvantages of each. (6)