

# T-106.4300 Web Software Development : Exam

Exam date: October 24th 2011, Lecture Hall T1

- All answers should easily fit on a single answer sheet.
- Write on each paper your name, your student number including any checking characters, the code and name of this course, examination hall, date, the number of papers returned and your signature.
- Return all papers, including scratch papers.
- If the question number is followed by an asterisk \*, you should select all options that apply. If there is no asterisk, select only one.
- All questions are worth one point. Some are easier, some are harder.
- No additional material allowed.

## Questions

### HTTP

1. Explain briefly the different (a-e) parts of the URL:  

```
http://www.example.org:8901/foo?bar=32
```

  
--a-- --b-- --c-- -d- --e--
2. Which of the following best describes the HTTP status code 404?
  - a. The server failed to fulfill an apparently valid request
  - b. The requested resource was not found
  - c. The request was received, understood, and successfully processed
  - d. The request requires user authentication
3. What is the essential difference between HTTP methods GET and POST?

### HTML & CSS

4. \* The following CSS style definition  

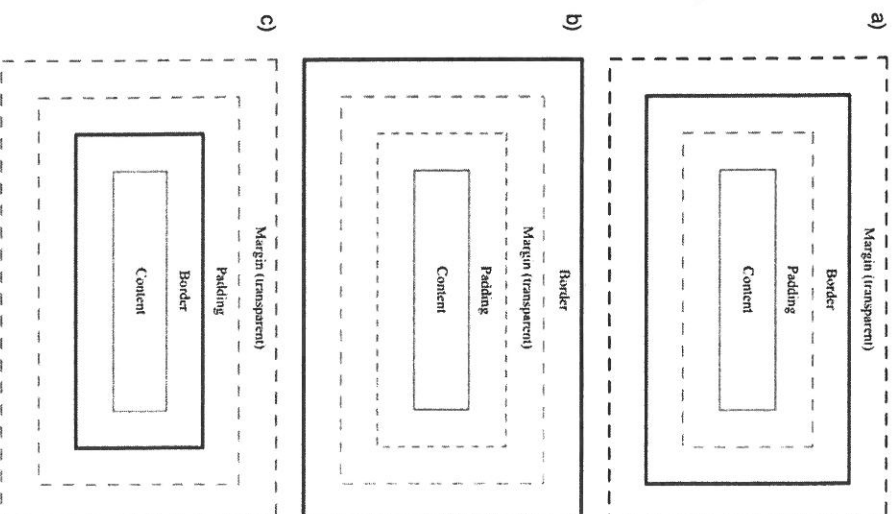
```
p strong {  
  font-weight: bold;  
}
```

  
makes the text bold in:
  - a. all paragraph elements
  - b. all paragraph elements that include a strong element
  - c. all strong elements that are anywhere inside any p element
  - d. all strong elements that are located immediately after any p element
5. \* Which statements about HTML elements are true?
  - a. h1, h2, ... h6 are meant for headings and subheadings in the text
  - b. for paragraphs of text, div element is semantically the best choice
  - c. external style sheets can be loaded using the link element
  - d. ul element is meant to add numbered lists on the page

6. Explain the following four lines of CSS style definition (max. 2 sentences per explained line)

```
h1.pagetitle {  
  font-family: Helvetica, Arial, sans-serif;  
  font-size: 120%;  
  border-bottom: 1px solid gray; }
```

7. Which of the following best describes the CSS box model?



8. For the following fragment of HTML

```
<div>
  <p id="first">Short paragraph of text.</p>
  
  <p id="second">More text.</p>
</div>
```

and CSS style definition

```
img { float: right; }
#first { float: left; }
#second { clear: both; }
```

Which of the following best describes the resulting layout?



### JavaScript and jQuery

9. Explain briefly what each element a-f does in the following piece of code. Don't explain the syntax but rather why the element is necessary

```
$("#myButton").click( function() { alert("Pop"); } );
```

10. What does the following JavaScript function do when given an array of numbers as a parameter? Explain briefly on a higher level instead of explaining each line.

```
function myFunction(arr) {
  var found = Number.MAX_VALUE;
  for (var i = arr.length; --i; ) {
    if (arr[i] < found) {
      found = arr[i];
    }
  }
  return found;
}
```

### Django

- 11. Which claims below are true about Django's Models
  - a. ForeignKey fields are used to refer to other objects.
  - b. Models convert data from relational database form to objects and vice versa
  - c. Models are typically changed using template tags
  - d. Django's models don't have to be designed for a specific relational database (MySQL, Postgres, etc.).

12. Which of the following statements are true for Django templates?

- a. In addition to HTML, templates can also render XML, JSON, and text
- b. Templates can inherit other templates
- c. Variables can be used in templates this way: {{variable\_name}}
- d. Templates can contain and execute arbitrary Python code
- e. Templates can specify any HTTP Headers for the response

13. If you have a model named ExamResult, how can you fetch all ExamResult objects from the database?

- a. ExamResult.objects.all()
- b. get\_all\_objects(ExamResult)
- c. ExamResult.all()
- d. ExamResult.objects.filter()

### REST & SOAP

14. Which of the following are true?

- a. The key elements in REST are the resources.
- b. If needed, RESTful web services can be implemented using GET and POST only
- c. RESTful web services must use HTML

### Security

15. Django ships with an application to authenticate your users. What is the default authentication scheme?

- a. OpenID
- b. Form-based login
- c. HTTP Digest Authentication
- d. HTTP Basic Authentication

## T-106.4300 Web Software Development : Exam

Exam date: December 13th 2011, Lecture Hall B

- All answers should easily fit on a single answer sheet.
- Write on each paper your name, your student number including any checking characters, the code and name of this course, examination hall, date, the number of papers returned and your signature.
- Return all papers, including scratch papers.
- If the question number is followed by an asterisk \*, you should select all options that apply. If there is no asterisk, select only one.
- All questions except 12 are worth one point. Some are easier, some are harder.
- No additional material allowed.

### Questions

#### HTML & CSS

1. For the following fragment of HTML

```
<div>
  <p id="first">Short paragraph of text...</p>
  
  <p id="second">More text...</p>
</div>
```

and CSS style definition

```
img { float: right; }
#first { float: left; }
#second { clear: both; }
```

Which of the following best describes the resulting layout?



2. The following CSS style definition

```
p i {
  font-style: italic;
}
```

makes the text italic in:

- all paragraph elements
- all paragraph elements that include an `i` element
- all `i` elements that are anywhere inside any `p` element
- all `i` elements that are located immediately after any `p` element

3. Explain briefly the different (a-e) parts of the URL:

```
http://www.example.org:801/fo2?bar=32
```

#### Python

4. What are the values of `dLength`, `dBreadth`, and `dArea` after the following Python code has been executed?

```
dLength = 0
dBreadth = 0
dArea = dLength * dBreadth
dLength = 5
dBreadth = 4
```

- 0, 0, 0
- 0, 0, 20
- 5, 4, 0
- 5, 4, 20

#### JavaScript & jQuery

5. Which of the following specify a function that will return a number?

- function myFunction() { return 3; }
- public Number function myFunction() { return 3; }
- var myFunction = function() { return 3; }
- Number function myFunction() { return 3; }

6. With jQuery, the easiest way to load new HTML content from URL `/products/2/` and replace element's (id `examplediv`) content is:

- `jQuery("#examplediv").update("/products/2/");`
- `jQuery("#examplediv").load("/products/2/");`
- `jQuery().get("/products/2/").update("#examplediv");`
- `jQuery().get("/products/2/").html("#examplediv");`

## Appendix 1

### urls.py

1. from django.conf.urls.defaults import patterns, include, url
2. urlpatterns = patterns('',
3. url(r'^polls/\$', 'polls.views.index'),
4. url(r'^polls/(?P<poll\_id>\d+)/\$', 'polls.views.detail'),
5. )

### polls/models.py

1. from django.db import models
- 2.
3. class Poll(models.Model):
4. question = models.CharField(max\_length=200)
5. pub\_date = models.DateTimeField('date published')
- 6.
7. class Choice(models.Model):
8. poll = models.ForeignKey(Poll)
9. choice = models.CharField(max\_length=200)
10. votes = models.IntegerField()

### polls/Views.py

1. from django.template import Context, loader
2. from django.shortcuts import render\_to\_response, get\_object\_or\_404
3. from django.http import HttpResponse
4. from polls.models import Poll
- 5.
6. def index(request):
7. latest\_poll\_list = Poll.objects.all().order\_by('-pub\_date')[:5]
8. t = loader.get\_template('index.html')
9. c = Context({'latest\_poll\_list': latest\_poll\_list })
10. return HttpResponse(t.render(c))
- 11.
12. def detail(request, poll\_id):
13. p = get\_object\_or\_404(Poll, pk=poll\_id)
14. return render\_to\_response('detail.html', {'poll': p})

### #index.html

1. {% if latest\_poll\_list %}
2. <ul>
3. {% for poll in latest\_poll\_list %}
4. <li><a href="/polls/TODOL/">TODOL</a></li>
5. {% endfor %}
6. </ul>
7. {% else %}
8. <p>No polls are available.</p>
9. {% endif %}

### #detail.html

1. <h1>{{ poll.question }}</h1>
2. <ul>
3. {% for choice in poll.choice\_set.all %}
4. <li>{{ choice.choice }}</li>
5. {% endfor %}
6. </ul>