HTML & CSS

1. What is the key difference between div and span elements? (max 2 sentences) (1 point)

2. What does the following CSS style definition make italic? (Max 1 sentence) (1 point)
   .p i {
     font-style: italic;
   }

HTTP

3) HTTP is a stateless protocol.
   a) What does it mean that the protocol is stateless? (1 point)
   b) How web services (e.g., REST APIs) benefit from HTTP being stateless? (1 point)

4) Consider the URL https://www.example.com/foobar?q=banana&id=561 Explain briefly what are the following parts of the URL (2 points)
   a) https://
   b) www.example.com
   c) /foobar
   d) ?q=banana&id=561
JavaScript & jQuery

5) The following JavaScript code is executed. What (if anything) is printed in the console? (2 points)

```javascript
function foo(x) {
    return function(y) {
        return x + y;
    };
}
var bar = foo(5);
console.log(bar(2));
```

6) Look at the code below. Explain what happens when you click the link. (2 points)

```html
<!doctype html>
<html>
<head>
<title>Demo</title>
</head>
<body>
<a href="http://jquery.com/">jQuery</a>
<script src="jquery.js"></script>
<script>
$( document ).ready(function() {
    $( "a" ).click(function( event ) {
        alert( "Hello" );
    });
});
</script>
</body>
</html>
```
APIs

7) You see a client sending the following requests to a REST API at example.com. Explain what you assume each request does. (one sentence per request, 3 points)
   
   a. GET http://www.example.com/customers/12345
   b. GET http://www.example.com/customers/12345/orders
   c. POST http://www.example.com/customers/12345/orders

Sessions & Security

8) Explain how CSRF works, and how CSRF tokens can be used to prevent this (3 points)

Django

9) (total 4 points)
   The following snippets are a Django view and the corresponding urls.py. The service hosts private photo albums for a number of users.

views.py

```python
@login_required(login_url='/login/')
def del_page(request, album, page_num):
    page = Page.objects.get_object_or_404(album=album, number=page_num)
    page.delete()
    return HttpResponseRedirect('albums/
```

urls.py

```python
urlpatterns = [
    url(r'^delete/(?P<album>[a-z]+)/(?P<page_num>[0-9]{2})$' views.del_page),
    ...other urls...
```

(3 points) Briefly explain different scenarios that can what happen when a user visits the URL: http://someserver.com/delete/vacation/42
Use the line numbers if necessary.

(1 point) There is an authorization problem in the code. What is it? (Max 2 sentences.)