

# Lab 2 solutions

## Exercise 1

### Question 1.

This will retrieve the content of the default webpage for vision.ucla.edu. The content type is text/html.

The content length is 6966 bytes.

The page was last modified on Monday, 7th March 2016 22:08:54 GMT (this time may change based on when you query the server).

The Accept-Range: bytes header field allows the server to send only a portion of an HTTP/1.1 message to the client. This is known as Byte serving. You can read up more about this [here](#).

### Question 2.

We don't receive the web page's content but just the header. The HEAD method is almost identical to GET. The main difference between them is that the web server will not return a message body in the response to a HEAD request. The meta-information contained in the HTTP headers in response to a HEAD request is identical to the information received in response to a corresponding GET request. The content type is text/html. The content length is 6966 bytes.

### Question 3.

You can do this by typing the following GET request (after establishing a telnet connection with vision.ucla.edu):

```
GET /people.html HTTP/1.1
```

```
host: www.ucla.edu.au
```

### Question 4.

The host header in the HTTP/1.1 request informs the server that the GET request is for the domain specified in the host. This is because, it is not uncommon that a single server (i.e. one IP address) can be multi-homed, i.e. the home of several Web domains. For example, "[www.abc.com](http://www.abc.com)" and "[www.abcd.com](http://www.abcd.com)" may both be hosted on the same server. So when this server receives a GET request, it must know which of these two domains the request corresponds to. This information is conveyed in the host header (e.g. host: [www.abc.com](http://www.abc.com)).

## Exercise 2

### Question 1.

The google webpage will set a cookie. This is evident from the following header received in response to the GET request.

Set-

Cookie:NID=77=plQOYkjtqkX4wPnfF6C1fo7IFcJ9cu2KvG7pZxVQ7PHGQcqni7OsKS9\_iQ2XnyEKmfDyPb

R52tpvo6gU-4rXbLsDNxKJaW6EJfR1hNZdyXEivjI8s6eVSycuLbAYIb; expires=Wed, 07-Sep-2016 00:55:33 GMT; path=/; domain=.google.com.au; HttpOnly

Examining the headers received in response to the GET request sent to the vision.ucla.edu page, does not reveal such a Set-Cookie header. Hence, it would appear this website does not set a cookie.

## Question 2.

I used FireFox for this exercise.

The google.com.au page sets two cookies. You can find more details about these cookies [here](#).

## Question 3.

The vision.ucla.edu sets five cookies. The reason why the cookies only show up during the browser session (and not during telnet as in Question 1) is two-fold. First of all, the GET request sent via telnet (as in Exercise 1) does not request any external resources such as images or embedded videos (e.g. if there are embedded youtube videos on a page, then youtube cookies would be stored - not the case in our example here). Secondly, telnet does not run any javascript code. If you examine the page source of the vision.ucla.edu. home page, you will notice that it contains a javascript tag that runs the following piece of code: <http://www.google-analytics.com/ga.js> which is responsible for setting the vision.ucla.edu cookies in your browser. The html page is referencing the javascript code in such a way that the origin of the cookies appears to be vision.ucla.edu and not google-analytics.com. You can find more details about the google analytics cookies [here](#).

# Exercise 3

## Question 1 (0.5 mark)

The status code is 200 and the phrase is OK.

## Question 2. (0.5 mark)

The file was last modified on Tuesday, 23rd September 2003 at 05:29 GMT. The response does indeed contain a DATE header, which indicates the time when the HTTP response message was actually generated by the server. Since these two fields are showing the same time in the current response, it is very likely that the server is simply setting the file's last modified time to the time when the HTTP response message is created.

## Question 3. (0.5 mark)

Both the client and server are configured for using persistent HTTP connections. Look at the Connection: header in both the request and response.

## Question 4. (0.5 mark)

73 bytes of HTML data is being returned to the browser.

## Question 5. (0.5 mark)

The file returned to the browser is an HTML file with the following content:

```
<html>
```

Congratulations. You've downloaded the file lab2-1.html!

</html>

## Exercise 4

### Question 1. (0.25 mark)

No, this header is not present in the request. This is because this is the first time that the browser is requesting this particular file or the local browser cache has just been emptied prior to requesting this file.

### Question 2. (0.25 mark)

The response does indicate that the file was last modified on Tuesday, 23rd September 2003 at 05:35 GMT.

### Question 3. (0.5 mark)

The second GET request does indeed contain the "If-Modified-Since" line and "If-None-Match" fields. The time specified here is Tuesday, 23rd September 2003 at 05:35 GMT which was obtained from the previous response message. The If-None-Match contains "1bfef-173-8f4ae900". In other words, the browser has cached the "LAST-MODIFIED" time from the previous response for the same page and is including that time in the subsequent response message. It is also asking server to check the Etag value with the previously received value in If-None-Match.

### Question 4. (0.75 mark)

The status code and phrase in the response are 304 Not Modified. This indicates to the browser that the server has not modified the page since the address specified in the "IF-MODIFIED-SINCE" time in the request. Hence, the server does not respond back with the requested file. The browser can simply display the locally cached version of this file.

### Question 5 . (0.75 mark)

What is the value of the Etag field in 2nd response message and how it is used? Have this value changed since the 1st response message was received?

The entity tag (Tag) value is "1bfef-173-8f4ae900". The Etag value has not changed since the first response showing that the content has not been modified and that the locally cached version of this file can be used. ETag is another mechanism used by HTTP for cache validation. It is used in conjunction with "If-None-Match" header field.

## Exercise 5 (5 marks)