University of Stirling

Computing Science and Mathematics

CSCU9A2 Tutorial 4 – Sample solutions Spring 2017

(for week starting 27 February)

1.

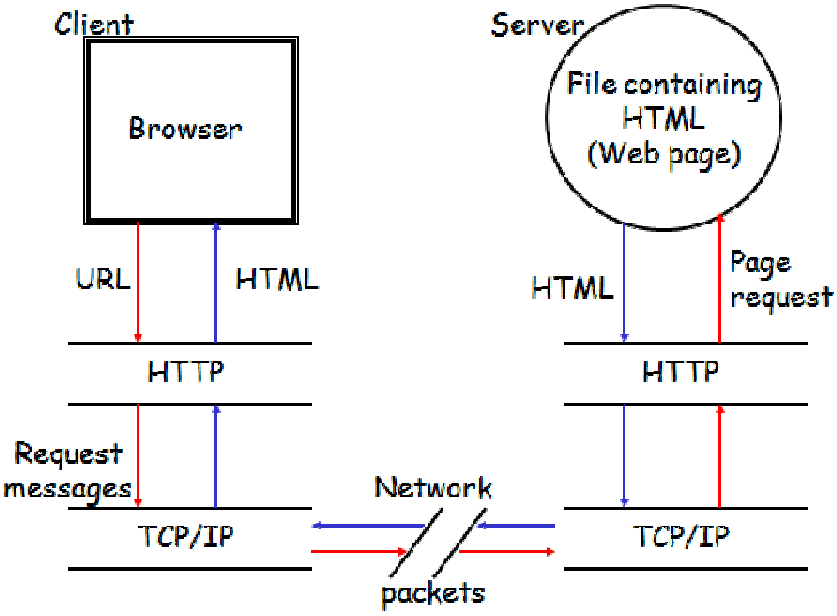
a. [Note: the wording of the question is a bit unclear. Let's say instead that you have typed in the URL `www.aserver.net/index.html` and the HTML shown is the content of the file identified by this URL.]

Refer to the picture below. HTTP handles communication between the client (the browser on the computer in 2A21) and the server.

The browser sends an HTTP request to the server at www.aserver.net asking for the file index.html. The request is sent over the internet (presumably) where TCP/IP is the underlying protocol communication protocol (we learned about these in CSC9A1). TCP/IP splits the message up into packets, which are then individually transmitted, and reassembled at the server.

The server looks up the file index.html and sends a copy of its content back to the client. The browser interprets the HTML in the file, displaying the title and the text in the paragraph. The image tag causes the browser to send a further request to the server, this time asking for the file pretty-picture.gif. The server returns a copy of this file, which is then displayed by the browser.

A key concept is that the browser does *not* "visit" the web page/site, but instead it gets sent *copies* of documents held on the server.



b. By using CSS, the content (text, images, links, etc) on a web page can be kept separate from the layout (colours, font sizes and styles, background images, etc). The layout is described using rules, which say how each element of the HTML should be styled. Typically, the content of the page is kept within a .html file and the CSS rules are kept within a separate .css file (though it is possible to have internal and inline stylesheets that are held within the same HTML file as the content).

The big benefit of using CSS is that allows the same style to be used consistently over all the pages within a website. CSS also makes it possible to easily switch between styles in an HTML document, simply by changing the stylesheet that is linked to in the head of the document. This feature can be used to provide alternative accessible versions of websites, suitable for users with certain visual or other impairments (such as dyslexia).

There are no real downsides to CSS. Not all of its features are supported in all browsers. There is a list of what is supported by various browsers at `http://www.w3schools.com/cssref/css3_browsersupport.asp`

2.

(a) `display(2);`

```
Enter display, 2 passed in, allocate new memory for parameter variable n and set to 2
    n is >0 so enter if branch:
        Output 2
        Call display(1)     (note:  "n-1" does not change n, just calculates a value)
        Enter display, 1 passed in, allocate new memory for parameter n and set to 1
            n is >0 so enter if branch:
                Output 1
                Call display(0)
                Enter display, 0 passed in, allocate new memory for parameter n and set to 0
                    n is ==0 so enter else branch:
                        Output 0
                    End of if
                End of display(0) call, return to suspended display(1)
                Output 1        (This call's memory for n still contains 1)
            End of if
        End of display(1) call, return to suspended display(2)
        Output 2        (This call's memory for n still contains 2)
    End of if
End of display(2) call, return and end of tracing
```

Overall, displays 2 1 0 1 2 (though in a column).

(b) Just displays `-1`

(c) The answer to (a) would not change.

The answer to (b) would change dramatically: An infinite recursion, counting down for ever and never reaching the point of counting up again: (Eventually the JVM would fail with a "stack overflow".)

-1 -2 -3 -4 -5 …

(d) New version could be

```
private void display(int n)
{
   if (n>0)
   {
      System.out.println(n);
      display(n-1);
      System.out.println(n);
   }
   else
   if (n<0)
   {
      System.out.println(n);
      display(n+1);
      System.out.println(n);
   }
   else
      System.out.println(n);
}
```

or

```
private void display(int n)
{
   if (n!=0)
   {
      System.out.println(n);
      if (n>0)
         display(n-1);
      else
         display(n+1);
      System.out.println(n);
   }
   else
      System.out.println(n);
}
```

or other solutions are possible…

SBJ /DEC February 2017