

## CIS3210 — Computer Networks

## Assignment 1

Due October 2<sup>nd</sup> 2009 in/before midnight

This assignment asks you to create an http client using the socket API and TCP. This client asks for two pages and displays parts of their contents.

Your course of action should follow this sequence (not mandatory; you may do it your own way instead):

1. A sample client is stored in file [here](#) (a small part of this working program was removed and you have to add it back to make it work). Retrieve it and decipher the request text. Make sure to understand the meaning of the various components of the request.
2. Add the socket manipulation code to this client and run it to see the server's response.
3. Modify this client so that it contacts [CBC](#) and asks for a random file. Run it and expect an error response.
4. Further modify your client so that it requests the (existing) [News](#) and examine the header of server's response (ignore the page itself for now). Repeat this step until the response is **200 OK**. Your request will close the connection with the server immediately after receiving the first part of the response.
5. Upgrade your client so that it accepts the whole response from [CBC](#) and prints in ASCII:
  - a The header of the response (up to the CRLF CRLF inclusive).
  - b A line made of 10 dots (.....).
  - c The trailer of the page which starts with the **html** command **</body>** up to the last byte in the transmitted packet. Print in ASCII (some characters may look funny or not print at all).
  - d A line made of 10 dots (.....).

You can view the source of the page (from [CBC](#)) with any browser by asking to display it called "[Page Source](#)" in [Firefox](#).

6. Add to your code a second request asking for: `money` (it exists).  
Print the header of the response and the trailer of the page in exactly the same way as for the `News` page.
7. You are done. E-mail your code as a single unzipped, text file to: `doobo@cis.uoguelph.ca` with a subject line similar to:  
**3210: assignment 1 submission**

## Deliverables

Your code will be written in C, C++ or Java. It must use the Socket API function calls directly without any wrappers (intermediate “methods” or similar). All socket traffic should be done by issuing system calls from the `send/receive` or `read/write` families (again, no wrappers).

You will submit on paper a printout of your code with a self-evaluation page attached (see appendix).

You will sign up for a demonstration of your code (the details will be discussed in class) and successfully demonstrate that your self assessment is correct.

## Grading

You will receive marks for reaching milestones as listed above. Reaching a milestone ranked higher implies having satisfied lower-marked ones.

If you got this far:	3	4	5a	5a+c	6
You get this much:	2	4	6	8	10

An additional bonus of 2 marks will be given to those who succeed in doing step 6 in persistent mode without closing and reopening the socket.

## Assignment 1

### Self evaluation form

Name: \_\_\_\_\_ Student ID# \_\_\_\_\_ Total:

Step	Done correctly	Not done	Other (explanation)
Code uses only send/rcv or read/write		If not, final grade = 0	
Paper copy submitted?		Try again	
3			
4			
5a+b			
5a+b+c+d			
6			
Bonus			