# CS348 Computer Networks
# Lab Excercises 3

*Indian Institute of Technology, Patna*
*January, 24, 2017*

**Instructions**: This is a demo based assignment that will require you to write codes for the server. You have to show the demo to the TA and submit the codes in a tgz file with name *assign3.tgz*. The submission date is 07.02.2017.

1. In this assignment, you will learn the basics of socket programming for TCP connections in Java: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet. You will also learn some basics of HTTP header format.

   You will need to develop a web server that handles one HTTP request at a time. Your web server should accept and parse the HTTP request, get the requested file from the servers file system, create an HTTP response message consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP 404 Not Found message back to the client.

   To run the web server, put an HTML file (e.g., HelloWorld.html) in the same directory that the server is in. Run the server program. Determine the IP address of the host that is running the server (e.g., 172.16.1.156). From another host, open a browser and provide the corresponding URL. For example: http://172.16.1.156:1234/HelloWorld.html
   HelloWorld.html is the name of the file you placed in the server directory. Note also the use of the port number after the colon. You need to replace this port number with whatever port you have used in the server code. In the above example, we have used the port number 1234. The browser should then display the contents of HelloWorld.html. If you omit ":1234", the browser will assume port 80 and you will get the web page from the server only if your server is listening at port 80. Then try to get a file that is not present at the server. You should get a 404 Not Found message.

   You will hand in the complete server code along with the screen shots of your client browser, verifying that you actually receive the contents of the HTML file from the server.

2. Extend the previous problem to implement a multithreaded web server that can open concurrent connections with more than one browsers. there should be a separate configuration file for the web server, that it should read initially to configure itself. The configuration file should have the following fields:

   (a) The maximum number of concurrent requests that the server would support.
   (b) The IP addresses that it would filter, i.e. the IP addresses that the web server would block.
   (c) The default html file along with its path name, that the webserver would return.