**Building a Simple Web Client and a Multithreaded Web Server**

**Client**

\*Client opens a socket connection with the server and requests for a file. Once it gets the file it logs the contents of the file in console

\*Takes server name , port number and the file names as parameter,forms the request and sends it to the server

\*reqFileName : Stores the requested file name

\*Client("localhost",port,reqFileName): Initiates the client object by takingthe localhost, port number and the file name as parameters

\*client\_Socket = new Socket(serverAddress, port): open a socket connection with server

\* client\_Socket.getInetAddress().getHostName(): log server name

\* client\_Socket.getRemoteSocketAddress(): ip address and port number

\* client\_Socket.getPort(): port number

\*requestFile(fileName):sends the request to the server

\* Client sends a GET request message to the server via the socket connection

\* Round Trip Time is calculated and logs the response message from the server

**Server**

\*Server class accepts requests from clients and gives the request to thread process for each incoming request so that it can handle multiple clients

\*Server sets the port number from the argument passed from command line

\*ServerSocket serverSocket = new ServerSocket(port): Listens to the socket connection

\*Server accepts http requests from the client all the time and creates a thread for each request to be handled

\* Socket clientSocket = serverSocket.accept():Listen for a TCP connection request and accept it whenever the clients requests for a file

\* HttpRequestResponse: This class process the http request and gives back the response accordingly

\* Calls processRequest methodfor processing the http request

\* StringTokenizer tokens = new StringTokenizer(requestLine): Extract the filename from the request line

\* If no file requested it fetches and displays the default page

\* Server sends the message body of the requested file to the client