**TIMECLIENT**

**import** java.io.\*;

**import** java.net.\*;

**public** **class** TimeClient {

**public** **static** **void** main(String[] args) **throws** IOException {

// Set up the socket, in and out variables

Socket TimeClientSocket = **null**;

PrintWriter out = **null**;

BufferedReader in = **null**;

**try** {

TimeClientSocket = **new** Socket("localhost", 4545);

out = **new** PrintWriter(TimeClientSocket.getOutputStream(), **true**);

in = **new** BufferedReader(**new** InputStreamReader(TimeClientSocket.getInputStream()));

} **catch** (UnknownHostException e) {

System.*err*.println("Don't know about host: localhost ");

System.*exit*(1);

} **catch** (IOException e) {

System.*err*.println("Couldn't get I/O for the connection to: 4545.");

System.*exit*(1);

}

BufferedReader stdIn = **new** BufferedReader(**new** InputStreamReader(System.*in*));

String fromServer;

String fromUser;

System.*out*.println("Initialised client and IO connections");

// This is modified as it's the client that speaks first

**while** (**true**) {

fromUser = stdIn.readLine();

**if** (fromUser != **null**) {

System.*out*.println("Client: " + fromUser);

out.println(fromUser);

}

fromServer = in.readLine();

System.*out*.println("Server: " + fromServer);

}

// Tidy up - not really needed due to true condition in while loop

// out.close();

// in.close();

// stdIn.close();

// TimeClientSocket.close();

}

}

**TIMESERVER**

**import** java.net.\*;

**import** java.io.\*;

**public** **class** TimeServer {

**public** **static** **void** main(String[] args) **throws** IOException {

ServerSocket timeServerSocket = **null**;

**boolean** listening = **true**;

// Make the server socket

**try** {

timeServerSocket = **new** ServerSocket(4545);

} **catch** (IOException e) {

System.*err*.println("Could not start TimeServer specified port");

System.*exit*(-1);

}

System.*out*.println("Server started");

**while** (listening){

**new** TimeServerThread(timeServerSocket.accept()).start();

System.*out*.println("New Time Server thread started");

}

timeServerSocket.close();

}

}

**TIMESERVERSTATE**

**import** java.net.\*;

**import** java.io.\*;

**import** java.util.Date;

**public** **class** TimeServerState {

//Set up the date object

Date date = **new** Date();

/\* The processInput method \*/

**public** String processInput(String theInput) {

System.*out*.println("processInput "+theInput);

String theOutput = **null**;

// Check what the client said

**if** (theInput.equalsIgnoreCase("What time is it?")) {

//Correct request

theOutput = date.toString();

} **else** { //incorrect request

theOutput = "Incorrect request - only understand \"What time is it?\"";

}

//Return the output message to the TimeServer

System.*out*.println(theOutput);

**return** theOutput;

}

}

**TIMESERVERTHREAD**

**import** java.net.\*;

**import** java.io.\*;

**public** **class** TimeServerThread **extends** Thread {

**private** Socket timeSocket = **null**;

**public** TimeServerThread(Socket timeSocket) {

**super**("TimeServerThread");

**this**.timeSocket = timeSocket;

}

**public** **void** run() {

**try** {

System.*out*.println("Initialising thread IO connections and state object");

PrintWriter out = **new** PrintWriter(timeSocket.getOutputStream(), **true**);

BufferedReader in = **new** BufferedReader(**new** InputStreamReader(timeSocket.getInputStream()));

String inputLine, outputLine;

System.*out*.println("Here");

TimeServerState tss = **new** TimeServerState();

System.*out*.println("Here1");

//This is the trick - in KK the server communicated first... Need to change so that the Client communicates first

**while** ((inputLine = in.readLine()) != **null**) {

System.*out*.println("Here2 " + inputLine);

outputLine = tss.processInput(inputLine);

out.println(outputLine);

**if** (outputLine.equals("Bye"))

**break**;

}

out.close();

in.close();

timeSocket.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}