Assignment 1

Socket Programming

Advanced Computer Networks

BY HRISHRAJ G L

Client side: HRISHIRAJ GL

Server Side: JITHIN K J

**SERVER SIDE CODE:**

|  |
| --- |
| package socket;  import java.io.\*;  import java.net.\*;  class Server {  public static void main(String args[])  throws Exception  {  // Create server Socket  ServerSocket ss = new ServerSocket(1234);  // connect it to client socket  Socket s = ss.accept();  System.out.println("Connection established");  // to send data to the client  PrintStream ps  = new PrintStream(s.getOutputStream());  // to read data coming from the client  BufferedReader br  = new BufferedReader(  new InputStreamReader(  s.getInputStream()));  // to read data from the keyboard  BufferedReader kb  = new BufferedReader(  new InputStreamReader(System.in));  // server executes continuously  while (true) {  String str, str1;  // repeat as long as the client  // does not send a null string  // read from client  while ((str = br.readLine()) != null) {  System.out.println(str);  str1 = kb.readLine();  // send to client  ps.println(str1);  }  // close connection  ps.close();  br.close();  kb.close();  ss.close();  s.close();  // terminate application  System.exit(0);  } // end of while  }  } |

**CLIENT SIDE CODE:**

|  |
| --- |
| package socket;  import java.io.\*;  import java.net.\*;  class Client {  public static void main(String args[])  throws Exception  {  // Create client socket  Socket s = new Socket("192.168.56.1",1234);  // to send data to the server  DataOutputStream dos  = new DataOutputStream(  s.getOutputStream());  // to read data coming from the server  BufferedReader br  = new BufferedReader(  new InputStreamReader(  s.getInputStream()));  // to read data from the keyboard  BufferedReader kb  = new BufferedReader(  new InputStreamReader(System.in));  String str, str1;  // repeat as long as exit  // is not typed at client  while (!(str = kb.readLine()).equals("exit")) {  // send to the server  dos.writeBytes(str + "\n");  // receive from the server  str1 = br.readLine();  System.out.println(str1);  }  // close connection.  dos.close();  br.close();  kb.close();  s.close();  }  } |

**OUTPUT :**

**Server side:**



**Client side**

