1. Implement a client-server program where the server sends a message and the client receives it and then the client sends a message and the server receives it, one time.

import java.io.\*;

import java.net.\*;

class Server

{

public static void main(String[] args)

{

try

{

System.out.println("Server Started");

ServerSocket srvr= new ServerSocket(1234);

Socket skt= srvr.accept();

System.out.println("Client-Server connection established\n");

BufferedReader brconsole= new BufferedReader(new InputStreamReader(System.in));

PrintWriter pwsocket= new PrintWriter(skt.getOutputStream(),true);

System.out.print("Enter your message: ");

String omsg= brconsole.readLine();

pwsocket.println(omsg);

pwsocket.flush();

BufferedReader brsocket= new BufferedReader(new InputStreamReader(skt.getInputStream()));

String imsg= brsocket.readLine();

System.out.print("Message received: ");

System.out.println(imsg);

brsocket.close();

pwsocket.close();

skt.close();

srvr.close();

}

catch (Exception e)

{

System.out.println("ERROR: "+e.getMessage());

}

}

}

class Client

{

public static void main(String[] args)

{

try

{

System.out.println("Client Started\n");

Socket skt= new Socket("localhost",1234);

BufferedReader brsocket= new BufferedReader(new InputStreamReader(skt.getInputStream()));

String imsg= brsocket.readLine();

System.out.print("Message received: ");

System.out.println(imsg);

BufferedReader brconsole= new BufferedReader(new InputStreamReader(System.in));

PrintWriter pwsocket= new PrintWriter(skt.getOutputStream(),true);

System.out.print("Enter your message: ");

String omsg= brconsole.readLine();

pwsocket.println(omsg);

pwsocket.flush();

pwsocket.close();

brsocket.close();

skt.close();

}

catch (Exception e)

{

System.out.println("ERROR: "+e.getMessage());

}

}

}

1. Implement a client-server program where the server sends a message and the client receives it and then the client sends a message and the server receives it, and the process continues till either enters a terminating string.

import java.io.\*;

import java.net.\*;

class Server

{

public static void main(String[] args)

{

try

{

System.out.println("Server Started <enter EXIT to terminate>");

ServerSocket srvr= new ServerSocket(1234);

Socket skt= srvr.accept();

System.out.println("Client-Server connection established\n");

BufferedReader brconsole= new BufferedReader(new InputStreamReader(System.in));

PrintWriter pwsocket= new PrintWriter(skt.getOutputStream(),true);

BufferedReader brsocket= new BufferedReader(new InputStreamReader(skt.getInputStream()));

String omsg,imsg;

while(true)

{

System.out.print("Enter your message: ");

omsg= brconsole.readLine();

pwsocket.println(omsg);

pwsocket.flush();

if(omsg.equalsIgnoreCase("exit"))

break;

System.out.print("Message received: ");

imsg= brsocket.readLine();

if(imsg.equalsIgnoreCase("exit"))

{

System.out.println("Bye.. Exiting chat.");

break;

}

System.out.println(imsg);

}

brsocket.close();

pwsocket.close();

skt.close();

srvr.close();

}

catch (Exception e)

{

System.out.println("ERROR: "+e.getMessage());

}

}

}

class Client

{

public static void main(String[] args)

{

try

{

System.out.println("Client Started <enter EXIT to terminate>\n");

Socket skt= new Socket("localhost",1234);

BufferedReader brsocket= new BufferedReader(new InputStreamReader(skt.getInputStream()));

BufferedReader brconsole= new BufferedReader(new InputStreamReader(System.in));

PrintWriter pwsocket= new PrintWriter(skt.getOutputStream(),true);

String imsg, omsg;

while (true)

{

System.out.print("Message received: ");

imsg= brsocket.readLine();

if(imsg.equalsIgnoreCase("exit"))

{

System.out.println("Bye.. Exiting chat.");

break;

}

System.out.println(imsg);

System.out.print("Enter your message: ");

omsg= brconsole.readLine();

pwsocket.println(omsg);

pwsocket.flush();

if(omsg.equalsIgnoreCase("exit"))

break;

}

pwsocket.close();

brsocket.close();

skt.close();

}

catch (Exception e)

{

System.out.println("ERROR: "+e.getMessage());

}

}

}