TCP Socket Programming

1. Write code for Client and save in GreetingClient.java

```
// File Name GreetingClient.java
import java.net.*;
import java.io.*;

public class GreetingClient {

   public static void main(String [] args) {
      String serverName = "localhost";//args[0];
      int port = 1422;//Integer.parseInt(args[1]);
      try {
         System.out.println("Connecting to : " + serverName + " on PORT : " +
port);
         Socket client = new Socket(serverName, port);

         System.out.println("Just connected to : " +
client.getRemoteSocketAddress());
         OutputStream outToServer = client.getOutputStream();
         DataOutputStream out = new DataOutputStream(outToServer);

         out.writeUTF("Hello from : " + client.getLocalSocketAddress());
         InputStream inFromServer = client.getInputStream();
         DataInputStream in = new DataInputStream(inFromServer);

         System.out.println("Server says : " + in.readUTF());
         client.close();
      }catch(IOException e) {
         e.printStackTrace();
      }
   }
}
```

2. Write code for Server Side and save in GreetingServer.java

```
// File Name GreetingServer.java
import java.net.*;
import java.io.*;

public class GreetingServer extends Thread {
   private ServerSocket serverSocket;

   public GreetingServer(int port) throws IOException {
      serverSocket = new ServerSocket(port);
      serverSocket.setSoTimeout(10000);
   }

   public void run() {
      while(true) {
         try {
            System.out.println("Waiting 10secs for client on port " +
               serverSocket.getLocalPort() + "...");
            Socket server = serverSocket.accept();
```

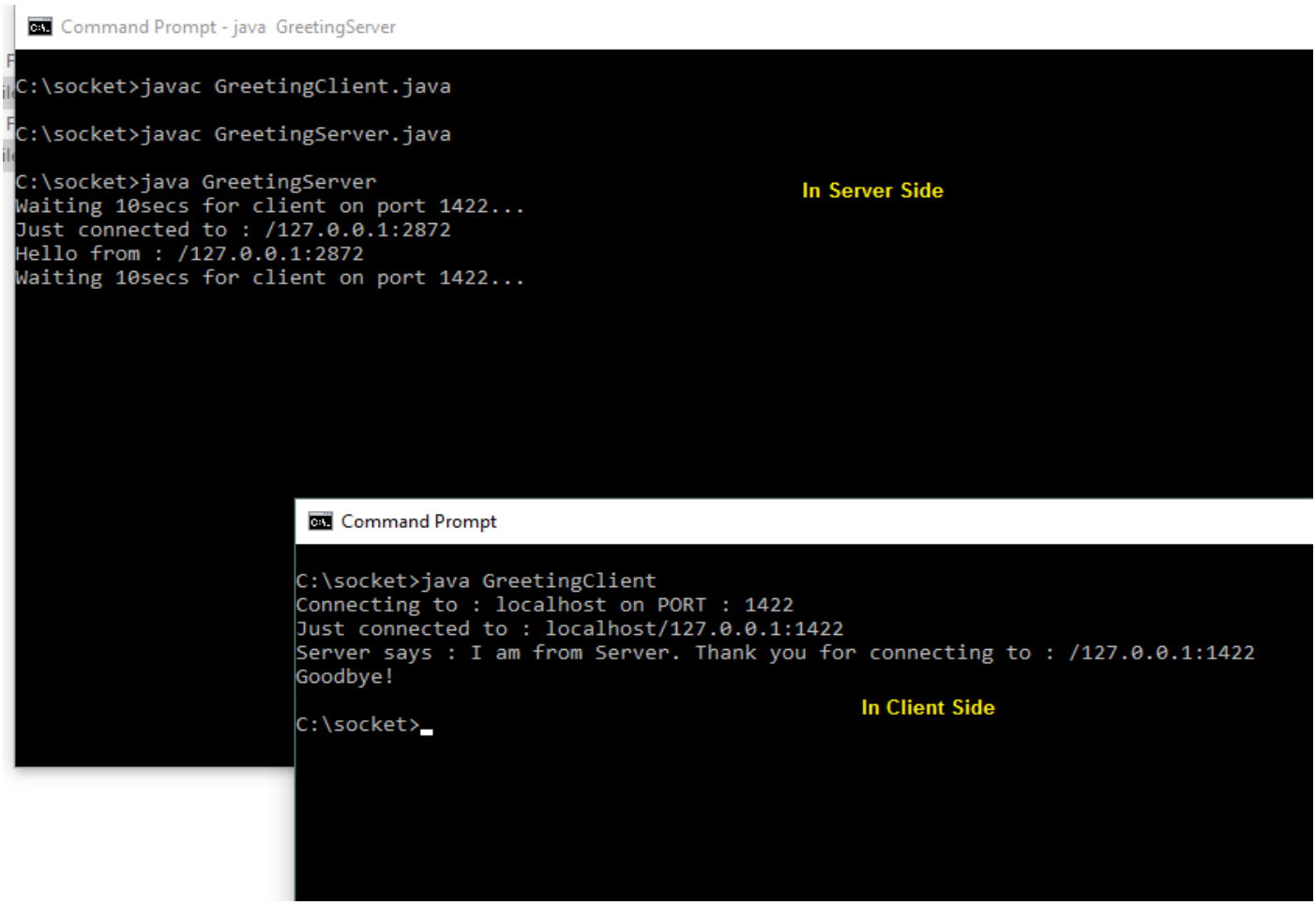
```
            System.out.println("Just connected to : " +
server.getRemoteSocketAddress());
            DataInputStream in = new DataInputStream(server.getInputStream());

            System.out.println(in.readUTF());
            DataOutputStream out = new
DataOutputStream(server.getOutputStream());
            out.writeUTF("I am from Server. Thank you for connecting to : " +
server.getLocalSocketAddress()
               + "\nGoodbye!");
            server.close();

         }catch(SocketTimeoutException s) {
            System.out.println("Socket timed out!");
            break;
         }catch(IOException e) {
            e.printStackTrace();
            break;
         }
      }
   }

   public static void main(String [] args) {
      int port = 1422;//Integer.parseInt(args[1]);
      try {
         Thread t = new GreetingServer(port);
         t.start();
      }catch(IOException e) {
         e.printStackTrace();
      }
   }
}
```

3. Make a project folder in C:\ e.g. socket
4. Open Command Prompt
5. Create directory as C:\socket>
6. Type C:\socket>javac GreetingServer.java and Enter
7. Type C:\socket>javac GreetingClient.java and Enter
8. Again open other Command Prompt for Server
9. Type C:\socket>java GreetingServer and Enter
10. In other command prompt for Client
11. Type C:\socket>java GreetingClient and Enter

NOTE : Run GreetingServer.java before GreetingClient.java

OUTPUT

1. TCP Socket Programming



UDP Socket Programming

1. Write code for Client and save in UDPClient.java

```
import java.io.*;
import java.net.*;

class UDPClient
{
   public static void main(String args[]) throws Exception
   {
      BufferedReader inFromUser =
         new BufferedReader(new InputStreamReader(System.in));
      DatagramSocket clientSocket = new DatagramSocket();
      InetAddress IPAddress = InetAddress.getByName("localhost");
      byte[] sendData = new byte[1024];
      byte[] receiveData = new byte[1024];
      String sentence = inFromUser.readLine();
      sendData = sentence.getBytes();
      DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
IPAddress, 9876);
      clientSocket.send(sendPacket);
      DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
```

```
        clientSocket.receive(receivePacket);
        String modifiedSentence = new String(receivePacket.getData());
        System.out.println("FROM SERVER:" + modifiedSentence);
        clientSocket.close();
    }
```

1. }**Write code for Server Side and save in UDPServer.java**

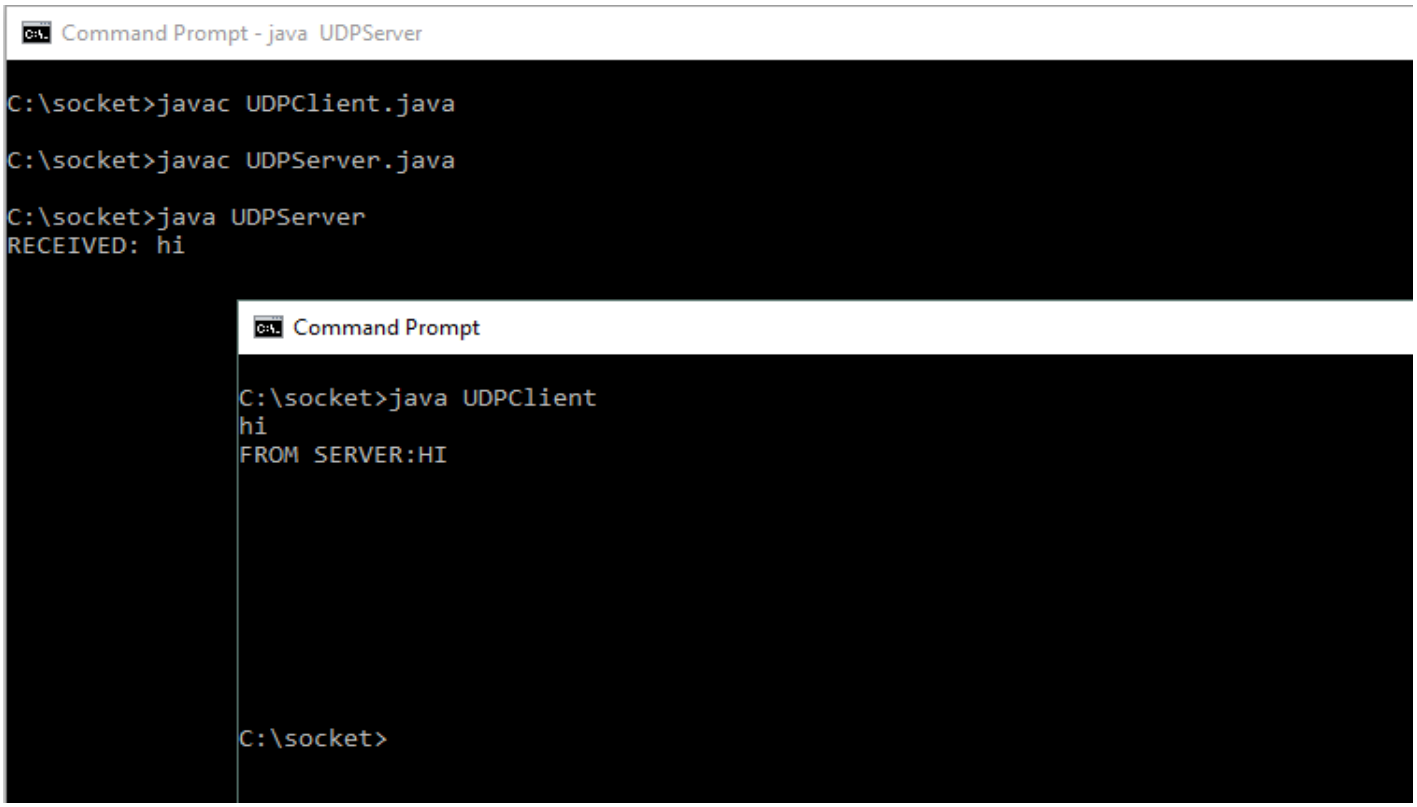
```
import java.io.*;
import java.net.*;

class UDPServer
{
   public static void main(String args[]) throws Exception
      {
         DatagramSocket serverSocket = new DatagramSocket(9876);
            byte[] receiveData = new byte[1024];
            byte[] sendData = new byte[1024];
            while(true)
               {
                  DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                  serverSocket.receive(receivePacket);
                  String sentence = new String( receivePacket.getData());
                  System.out.println("RECEIVED: " + sentence);
                  InetAddress IPAddress = receivePacket.getAddress();
                  int port = receivePacket.getPort();
                  String capitalizedSentence = sentence.toUpperCase();
                  sendData = capitalizedSentence.getBytes();
                  DatagramPacket sendPacket =
                  new DatagramPacket(sendData, sendData.length, IPAddress, port);
                  serverSocket.send(sendPacket);
               }
      }
```

1. }Make a project folder in C:\ e.g. socket
2. Open Command Prompt
3. Create directory as C:\socket>
4. Type C:\socket>javac UDPServer.java and Enter
5. Type C:\socket>javac UDPClient.java and Enter
6. Again open other Command Prompt for Server
7. Type C:\socket>java UDPServer and Enter
8. In other command prompt for Client
9. Type C:\socket>java UDPClient and Enter

   NOTE : Run UDPServer.java before UDPClient.java
10. Type hi and Enter

OUTPUT

2. UDP Socket Programming


Command Prompt - java UDPServer
C:\socket>javac UDPClient.java
C:\socket>javac UDPServer.java
C:\socket>java UDPServer
RECEIVED: hi
Command Prompt
C:\socket>java UDPClient
hi
FROM SERVER:HI
C:\socket>