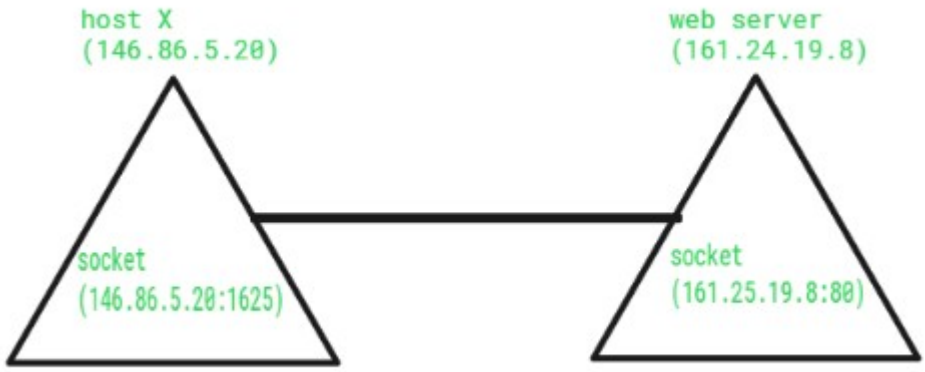
Title: Use Socket programming to create Client and Server to send Hello message using TCP

Socket

A socket is one endpoint of a two way communication link between two programs running on the network. The socket mechanism provides a means of inter-process communication (IPC) by establishing named contact points between which the communication take place.

Like ‘Pipe’ is used to create pipes and sockets is created using ‘socket’ system call. The socket provides bidirectional FIFO Communication facility over the network. A socket connecting to the network is created at each end of the communication. Each socket has a specific address. This address is composed of an IP address and a port number.

Socket are generally employed in client server applications. The server creates a socket, attaches it to a network port addresses then waits for the client to contact it. The client creates a socket and then attempts to connect to the server socket. When the connection is established, transfer of data takes place.

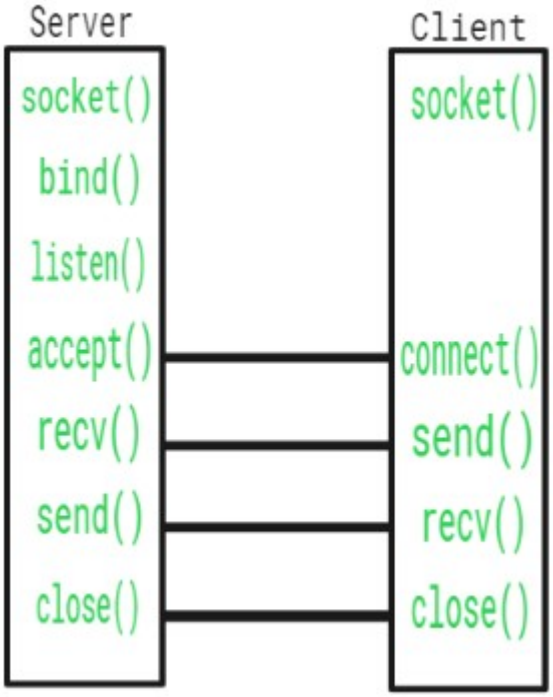


Types

1. **Datagram Socket :** This is a type of network which has connection less point for sending and receiving packets. It is similar to mailbox. The letters (data) posted into the box are collected and delivered (transmitted) to a letterbox (receiving socket).
2. **Stream Socket :** In Computer operating system, a stream socket is type of interprocess communications socket or network socket which provides a connection-oriented, sequenced, and unique flow of data without record boundaries with well defined mechanisms for creating and destroying connections and for detecting errors. It is similar to phone. A connection is established between the phones (two ends) and a conversation (transfer of data) takes place.

TCP (Transmission Control Protocol) Socket:

- TCP is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data.
- TCP sockets are commonly used for applications where data integrity and order are crucial, such as file transfers, email, and web browsing.
- A TCP socket connection is established through a three-way handshake, which involves the exchange of SYN (synchronize), SYN-ACK (synchronize-acknowledgment), and ACK (acknowledgment) packets between the communicating parties.
- Once the connection is established, data can be transmitted in both directions, and the protocol ensures that the data is delivered without errors and in the correct order.



Function Call	Description
Socket()	To create a socket
Bind()	It’s a socket identification like a telephone number to contact
Listen()	Ready to receive a connection
Connect()	Ready to act as a sender
Accept()	Confirmation, it is like accepting to receive a call from a sender
Write()	To send data
Read()	To receive data
Close()	To close a connection

Server.java

```
// A Java program for a Server
import java.net.*;
import java.io.*;
import java.util.*;

public class Server
{
//initialize socket and input stream
private Socket socket = null;
private ServerSocket server = null;
private DataInputStream in = null;
private DataOutputStream out=null;
Scanner sc=new Scanner(System.in);

// constructor with port
public Server(int port)
{
// starts server and waits for a connection
try
{
server = new ServerSocket(port);
System.out.println("Server started");
```

```
System.out.println("Waiting for a client ...");

socket = server.accept();
System.out.println("Client accepted");

// takes input from the client socket

in = new DataInputStream(
new BufferedInputStream(socket.getInputStream()));
out = new DataOutputStream(
socket.getOutputStream());
String line = "";

// reads message from client until "Over" is sent
while (!line.equals("Close"))
{
try
{
line = in.readUTF();
System.out.print("\nClient: "+line);
System.out.print("\nEnter Your msg: ");
line=sc.nextLine();
out.writeUTF("\n"+line);


}
catch(IOException i)
{
System.out.println(i);
}
}
System.out.println("Closing connection");

// close connection
socket.close();
in.close();
}
catch(IOException i)
{
System.out.println(i);
}
}

public static void main(String args[])
{
Server server = new Server(5000);
```

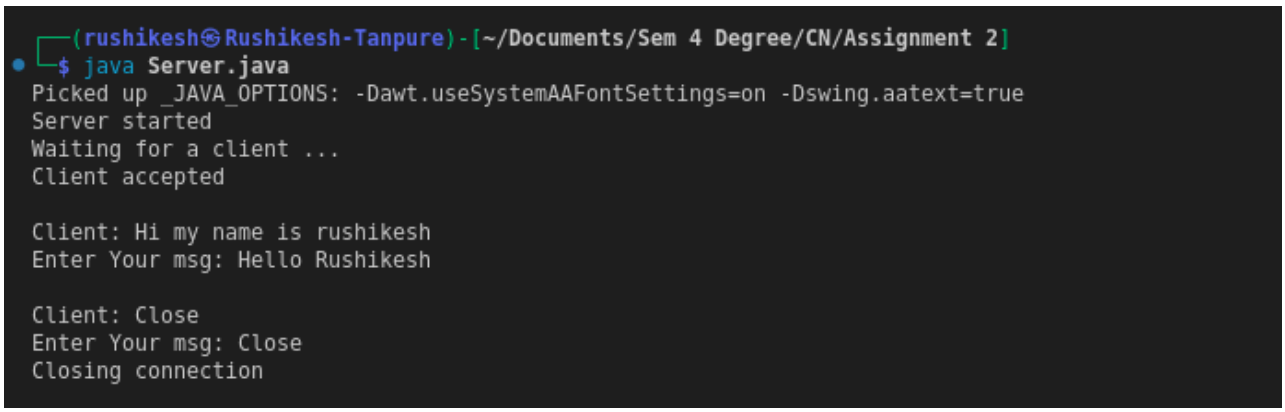
```
}
}
```

Client.java

```
// A Java program for a Client
import java.io.*;
import java.net.*;

public class Client {
// initialize socket and input output streams
private Socket socket = null;
private DataInputStream input,in = null;
private DataOutputStream out = null;

// constructor to put ip address and port
public Client(String address, int port)
{
// establish a connection
try {
socket = new Socket(address, port);
System.out.println("Connected");

// takes input from terminal
input = new DataInputStream(System.in);
in = new DataInputStream(
new BufferedInputStream(socket.getInputStream()));
// sends output to the socket
out = new DataOutputStream(
socket.getOutputStream());
}
catch (UnknownHostException u) {
System.out.println(u);
return;
}
catch (IOException i) {
System.out.println(i);
return;
}

// string to read message from input
String line = "";

// keep reading until "Over" is input
```

```
while (!line.equals("Close")) {
try {
System.out.print("\nEnter Your Msg: ");
line = input.readLine();
out.writeUTF(line);
line=in.readUTF();
System.out.print("Server: "+line);
}
catch (IOException i) {
System.out.println(i);
}
}

// close the connection
try {
input.close();
out.close();
socket.close();
}
catch (IOException i) {
System.out.println(i);
}
}

public static void main(String args[])
{
Client client = new Client("127.0.0.1", 5000);
}
}
```

Output:



```
┌──(rushikesh㉿Rushikesh-Tanpure)-[~/Documents/Sem 4 Degree/CN/Assignment 2]
└─$ java Client.java
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Note: Client.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
Connected

Enter Your Msg: Hi my name is rushikesh
Server: Hello Rushikesh
Enter Your Msg: Close
Server: Close
```