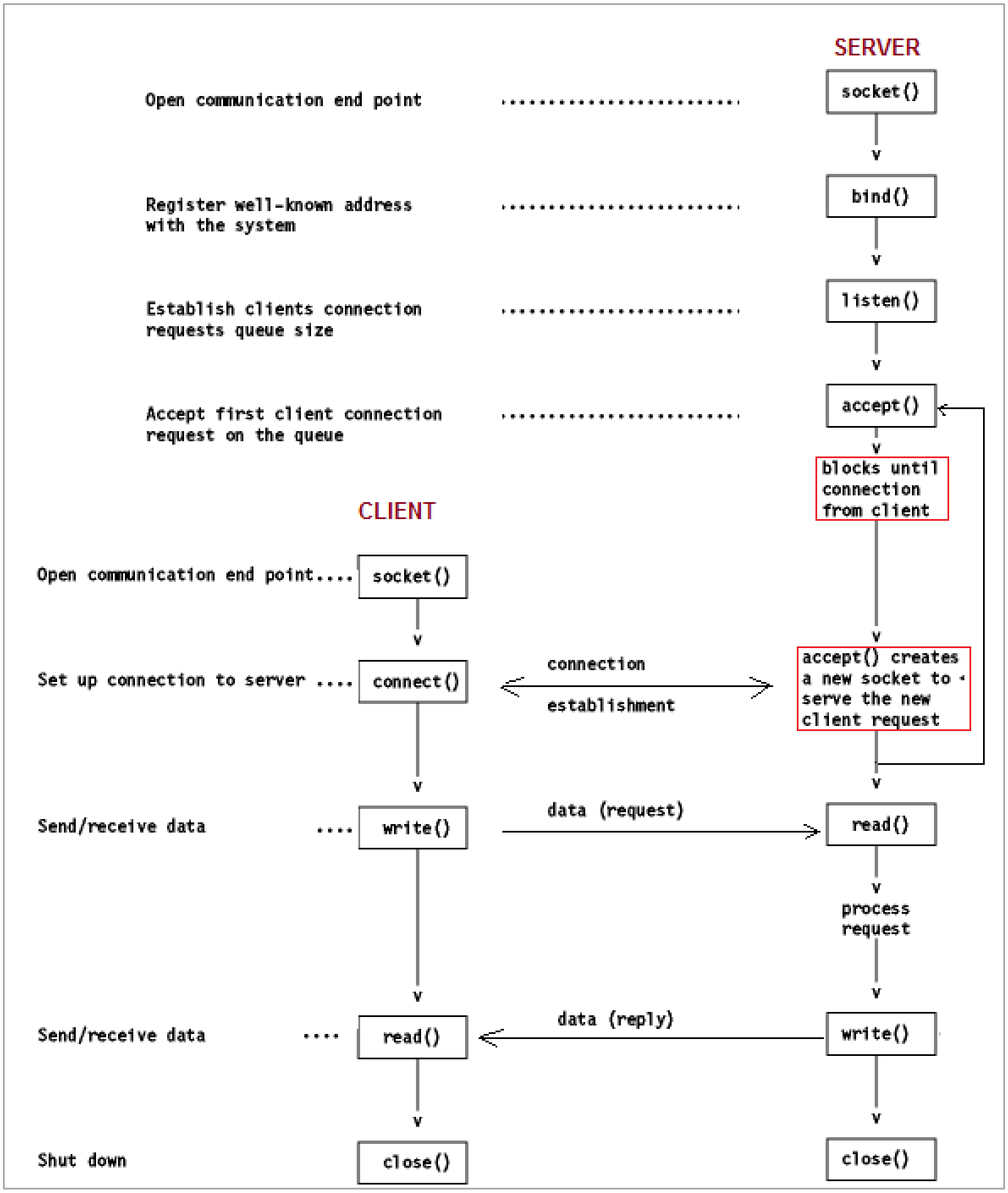

SERVER
Open communication end point
socket()
Register well-known address with the system
bind()
Establish clients connection requests queue size
listen()
Accept first client connection request on the queue
accept()
blocks until connection from client
CLIENT
Open communication end point....
socket()
Set up connection to server ....
connect()
connection
establishment
accept() creates a new socket to serve the new client request
Send/receive data ....
write()
data (request)
read()
process request
Send/receive data ....
read()
data (reply)
write()
Shut down
close()
close()

1. **Write code for Client and save in ChatClient.java**

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

public class ChatClient
{
     public static void main(String arg[])
     {
          try{
               Socket s=new Socket("Localhost",8888); // Socket()
               System.out.println("Server Connected : "+s); // Connect()

/ *** DataInputStream writes formatted binary data. DataOutputStream reads formatted
binary data. ***/
```

```
            DataInputStream din=new DataInputStream(s.getInputStream()); // Read ()
            DataOutputStream dout=new DataOutputStream(s.getOutputStream()); // Write

/*** BufferedReader reads text from a character-input stream, buffering characters so as
to provide for the efficient reading of characters, arrays, and lines ***/
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            String s1;
            do{
                s1=br.readLine();
                dout.writeUTF(s1);
                dout.flush();
                System.out.println("Server Message: I am from server.
"+din.readUTF());
            }while(!s1.equals("stop"));

        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

2. **Write code for Server Side and save in ChatServer.java**

Server

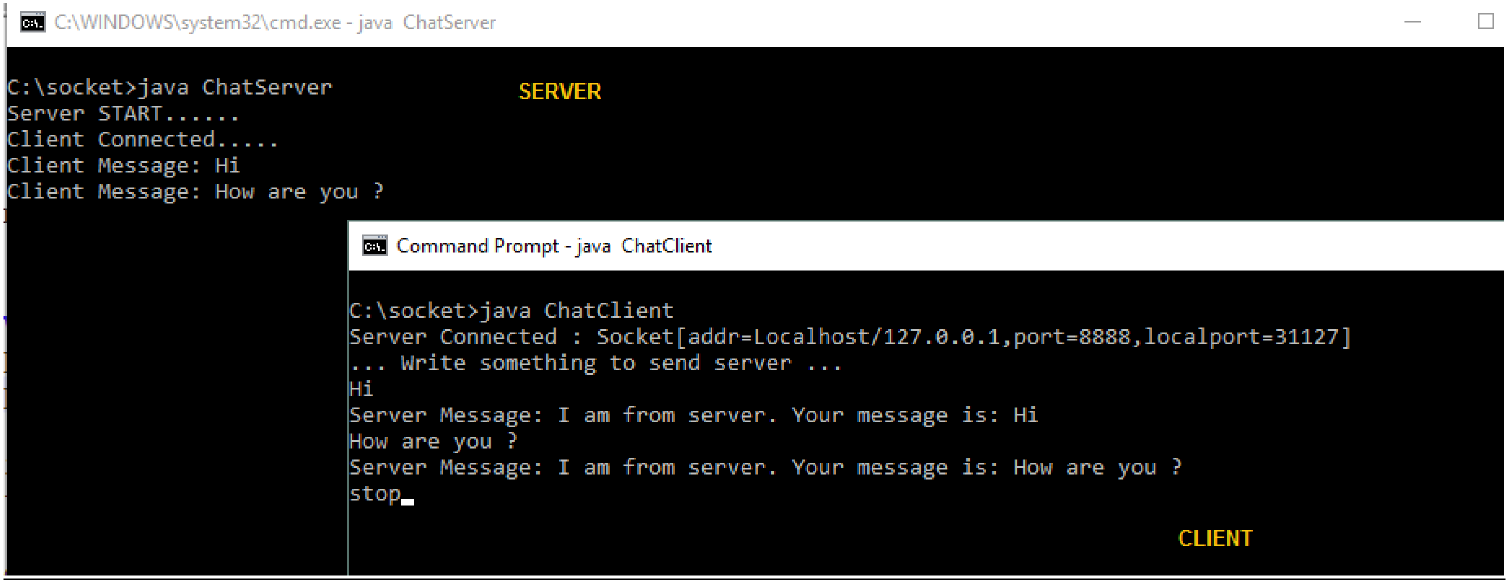
```
import java.io.*;
import java.net.*;
public class ChatServer
{
    public static void main(String arg[])
    {
        try{
            System.out.println("Server START......");
            ServerSocket ss=new ServerSocket(8888); // Socket()  and Bind() port
            Socket s=ss.accept(); // listen() and accept()
            System.out.println("Client Connected.....");
            DataInputStream din=new DataInputStream(s.getInputStream());// read()
            DataOutputStream dout=new DataOutputStream(s.getOutputStream());// write()

            String str=" ";
            do{
            str=din.readUTF();
            System.out.println("Client Message: "+str);
            dout.writeUTF("Your message is: "+str);
            dout.flush();

            }while(!str.equals("stop"));

        }catch(Exception e){System.out.println(e);}
    }

}
```

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 ChatServer.java and Enter
7. Type C:\socket>javac ChatClient.java and Enter
8. Again open other Command Prompt for Server
9. Type C:\socket>java ChatServer and Enter
10. In other command prompt for Client
11. Type C:\socket>java ChatClient and Enter

NOTE : Run GreetingServer.java before GreetingClient.java

OUTPUT


C:\WINDOWS\system32\cmd.exe - java ChatServer
SERVER
Command Prompt - java ChatClient
CLIENT

```
C:\socket>java ChatServer
Server START......
Client Connected.....
Client Message: Hi
Client Message: How are you ?
```

```
C:\socket>java ChatClient
Server Connected : Socket[addr=Localhost/127.0.0.1,port=8888,localport=31127]
... Write something to send server ...
Hi
Server Message: I am from server. Your message is: Hi
How are you ?
Server Message: I am from server. Your message is: How are you ?
stop
```