**Unit-1**

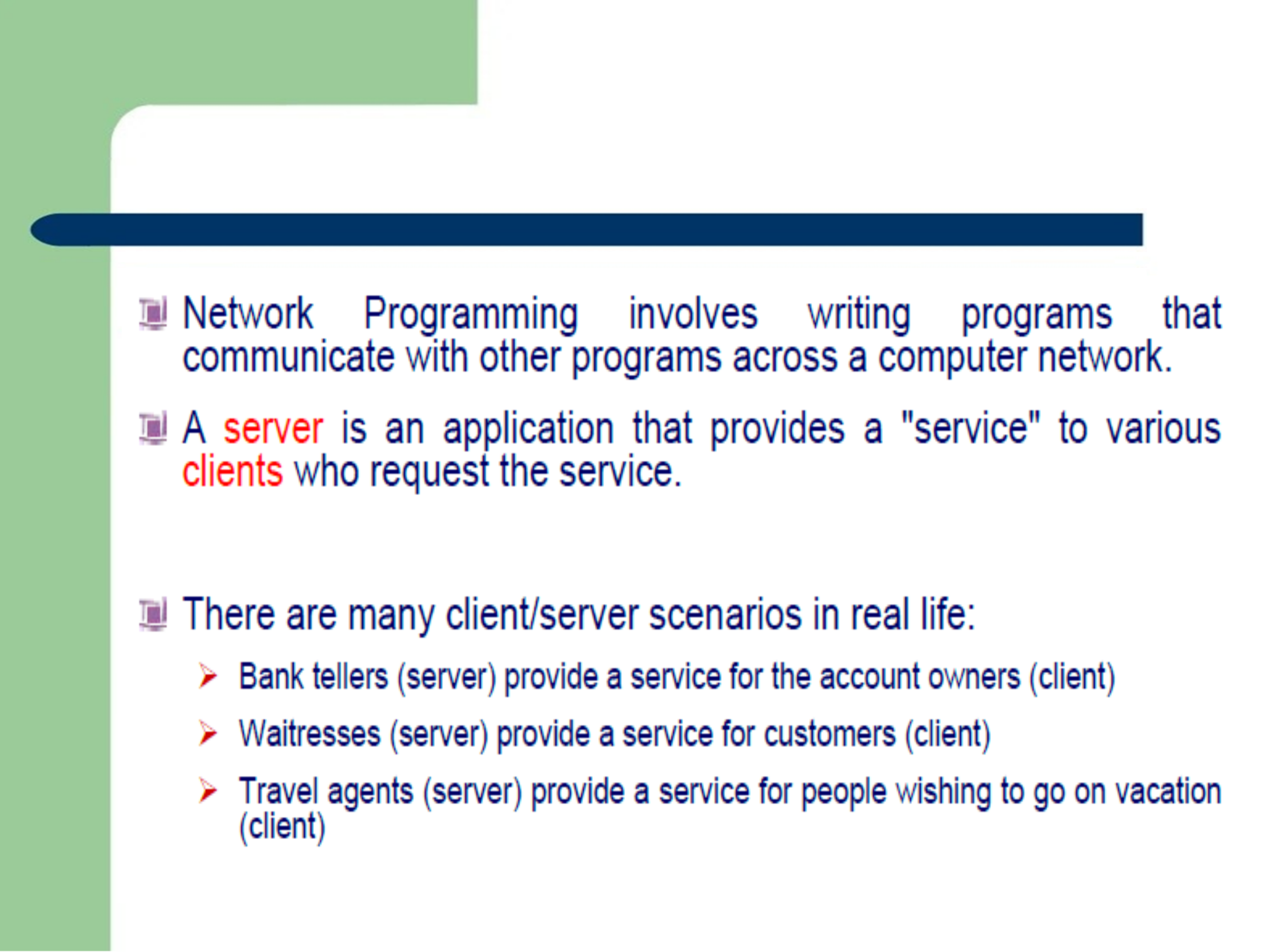
**Introduction**

1.1.Network Programming Features and Scope

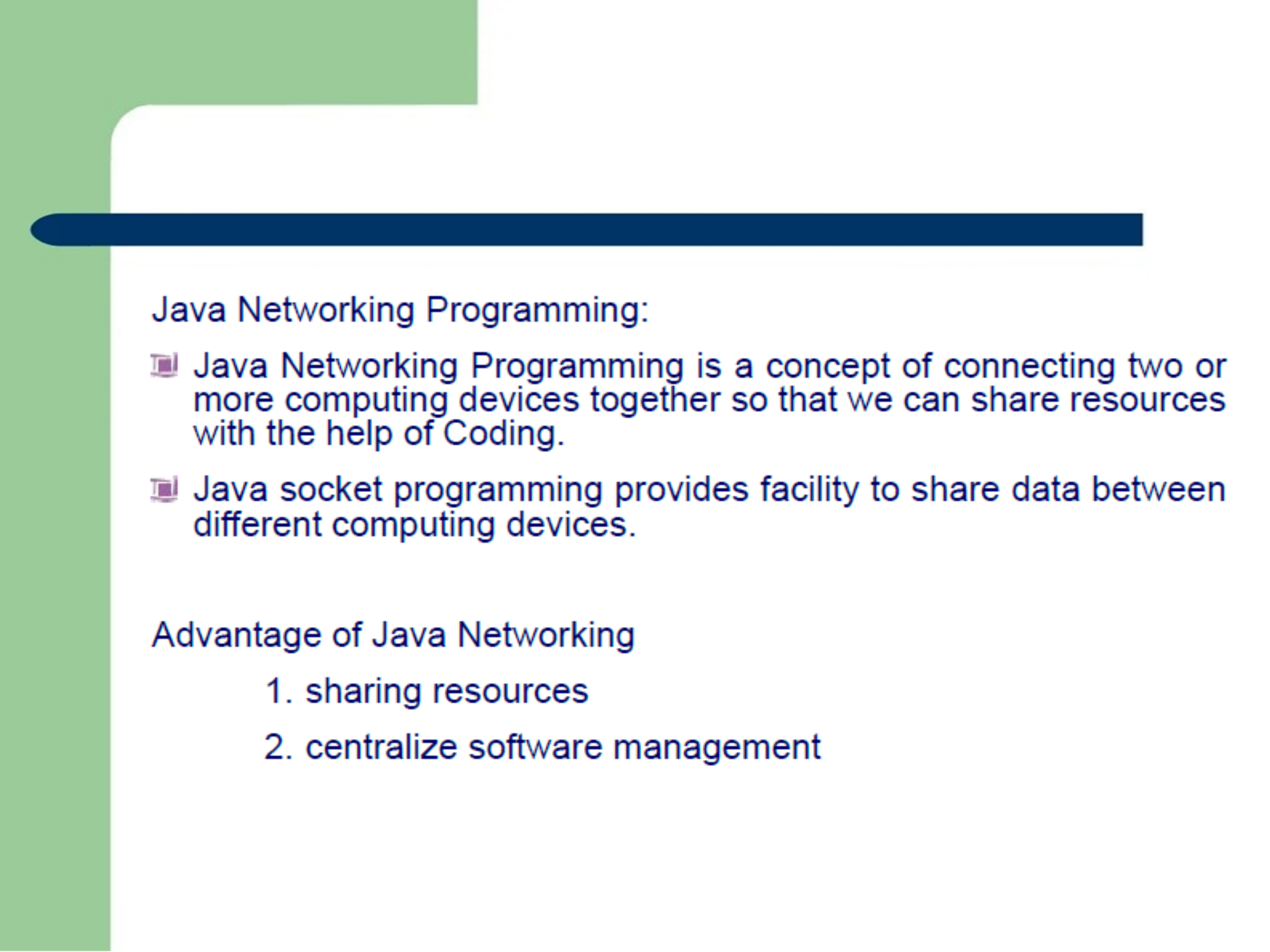
1.2.Network Programming Language, Tools & Platforms

1.3.Client and Server Applications

1.4.Client Server model and Software Design

**** **Unit-1**

**Introduction**

**** **Unit-1**

**Introduction**

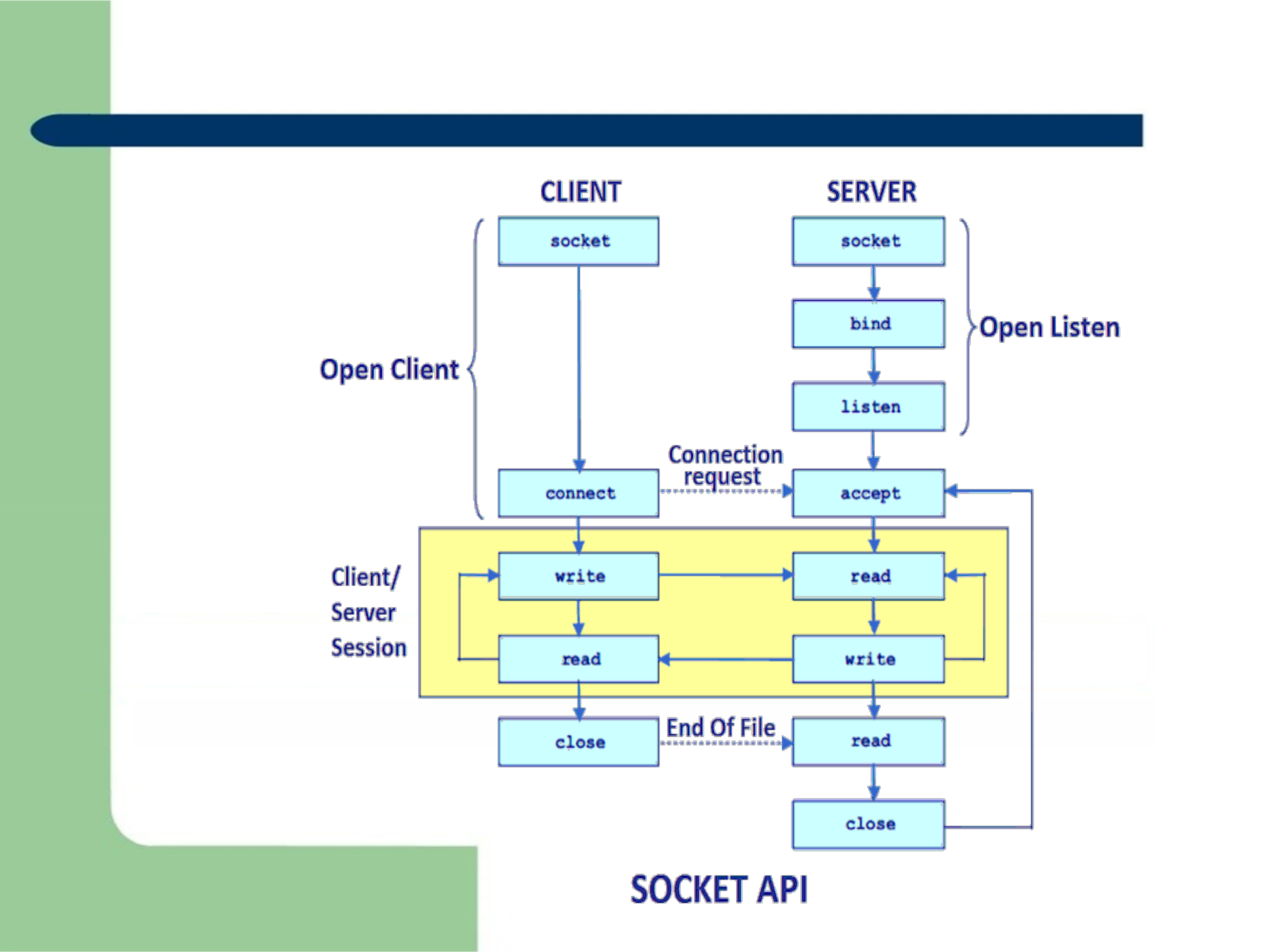
**1.3.Client and Server Applications**

* Java Socket programming is used for communication between the applications running on different JRE.
* Java Socket programming can be connection-oriented or connection-less.
* Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket

classes are used for connection-less socket programming.

***The client in socket programming must know two information:***

1. **IP Address of Server, and**
2. **Port number.**

**1.3.Client and Server Applications**



**1.3.Client and Server Applications**

**File: MyServer.java**

**import java.io.\*;**

**import java.net.\*;**

**public class MyServer {**

**public static void main(String[] args){ try{**

**File: MyClient.java**

**import java.io.\*;**

**import java.net.\*;**

**public class MyClient {**

**public static void main(String[] args) { try{**

|  |  |
| --- | --- |
| **ServerSocket ss=new ServerSocket(6666);** | **Socket s=new Socket("localhost",6666);** |
| **Socket s=ss.accept();//establishes connection** | **DataOutputStream dout=new DataOutputStream(s.getOut** |
| **DataInputStream dis=new DataInputStream(s.get** | **putStream());** |
| **InputStream());** | **dout.writeUTF("Hello Server");** |
| **String str=(String)dis.readUTF(); //return utf to** | **dout.flush();** |
| **String** | **dout.close();** |
| **System.out.println("message= "+str);** | **s.close();** |
| **ss.close();** | **}catch(Exception e)** |
| **}catch(Exception e)** | **{System.out.println(e);}** |
| **{System.out.println(e);}** | **}** |
| **}** | **}** |
| **}** |  |

***To execute this program open two command prompts and execute each program at each command prompt.***

***After running the client application, a message will be***

***displayed on the server console. UTF-****Stands for "Unicode*

*Transformation Format.*

**1.3.Client and Server Applications**

For example,

Among the **Constructors** for the *client-side* socket are the following:

Socket (InetAddress, int) — creates a socket and connects it to the specified port on the host at the specified IP address. Socket (String, int) — creates a socket and connects it to the specified port on the host named in String.

***Constructors*** on the *server side* include the following: ServerSocket (int) — creates a server socket and binds it to the specified port on the local host.

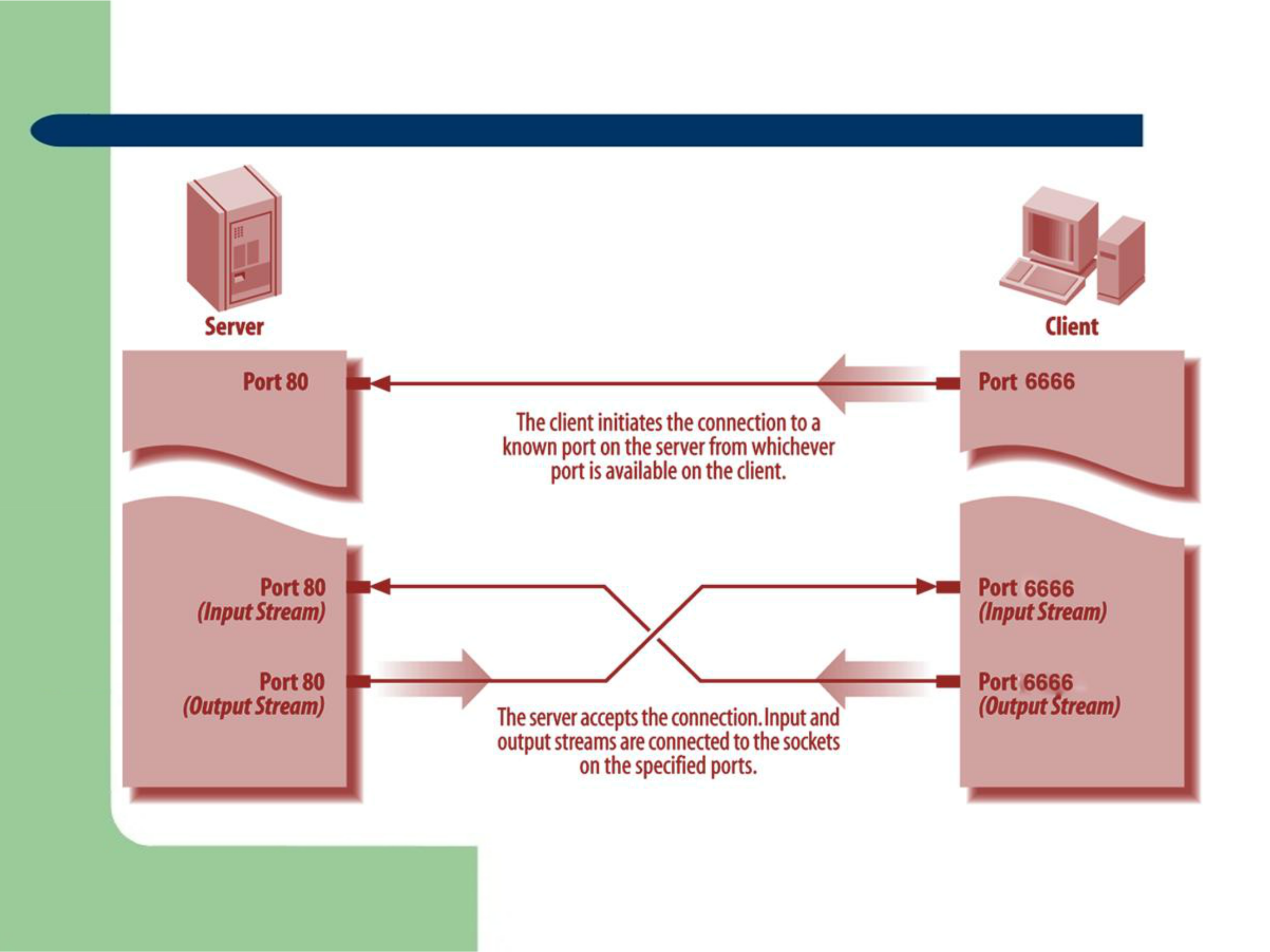
**Do You Know ?**

How to perform connection-oriented Socket Programming in networking ?

How to display the data of any online web page ?

How to get the IP address of any host name e.g. www.google.com ?

How to perform connection-less socket programming in networking ?

**1.3.Client and Server Applications**

***Figure . A client/server connection***

**Java Networking Terminology**

**The widely used java networking terminologies are given below:**

1. **IP Address**
2. **Protocol**
3. **Port Number**
4. **MAC Address**
5. **Connection-Oriented And Connection-Less Protocol**
6. **Socket**

**Java Networking Terminology**

**1) IP Address**

**IP address is a unique number assigned to a node of a network e.g. 192.168.0.1 . It is composed of range from 0 to 255. It is a logical address that can be changed.**

**2) Protocol**

**A protocol is a set of rules basically that is followed for communication. For example:**

**• http • TCP • FTP • Telnet • SMTP • POP etc.**

**3) Port Number**

**The port number is used to uniquely identify different applications. It acts as a communication endpoint between applications.**

**The port number is associated with the IP address for communication between two applications.**

**Java Networking Terminology**

**4) MAC Address**

**MAC (Media Access Control) Address is a unique identifier of NIC (Network Interface Controller). A network node can have multiple NIC but each with unique MAC.**

**5) Connection-Oriented And Connection-Less Protocol**

**In Connection-Oriented Protocol, acknowledgement is sent by the receiver. So it is reliable but slow. The example of connection-oriented protocol is TCP.**

**But, in Connection-Less Protocol, acknowledgement is not sent by the receiver. So it is not reliable but fast. The example of connection-less protocol is UDP.**

**6) Socket**

**A socket is an endpoint between two way communication.**