Name : Ankita Sanjay Kakade

PRN:12210626

Roll No.: 19

Subject: Computer Network Lab

**Assignment No.2**

1. Socket programming Report
2. What is socket: Socket is an end point of 2-way communication between programs running on the network.
3. Types of socket:

a) Stream socket (TCP) :

* + It is Reliable & Error free service
  + Sequential Packet delivered
  + Application: Telnet/SSH,HTTP,HTTPS,etc.
  + Conection-oriented Socket
  + Sequenced Packet Socket associate with Stream socket

b) Datagram Socket(UDP)

* Unreliable &best effort service
* Packet may arrive out of order
* Applications: Live Streaming audio/video
* Connectionless-oriented socket
* Raw Socket Associate with Datagram socket

1. What is Socket Programming: Socket Programming is a way of connecting two node on a network to communicate with each other

* Main functions in <sys/socket.h>are:

socket(): end point for communication -like a telephonne

bind(): assign Socket address

connect(): client connect to server port

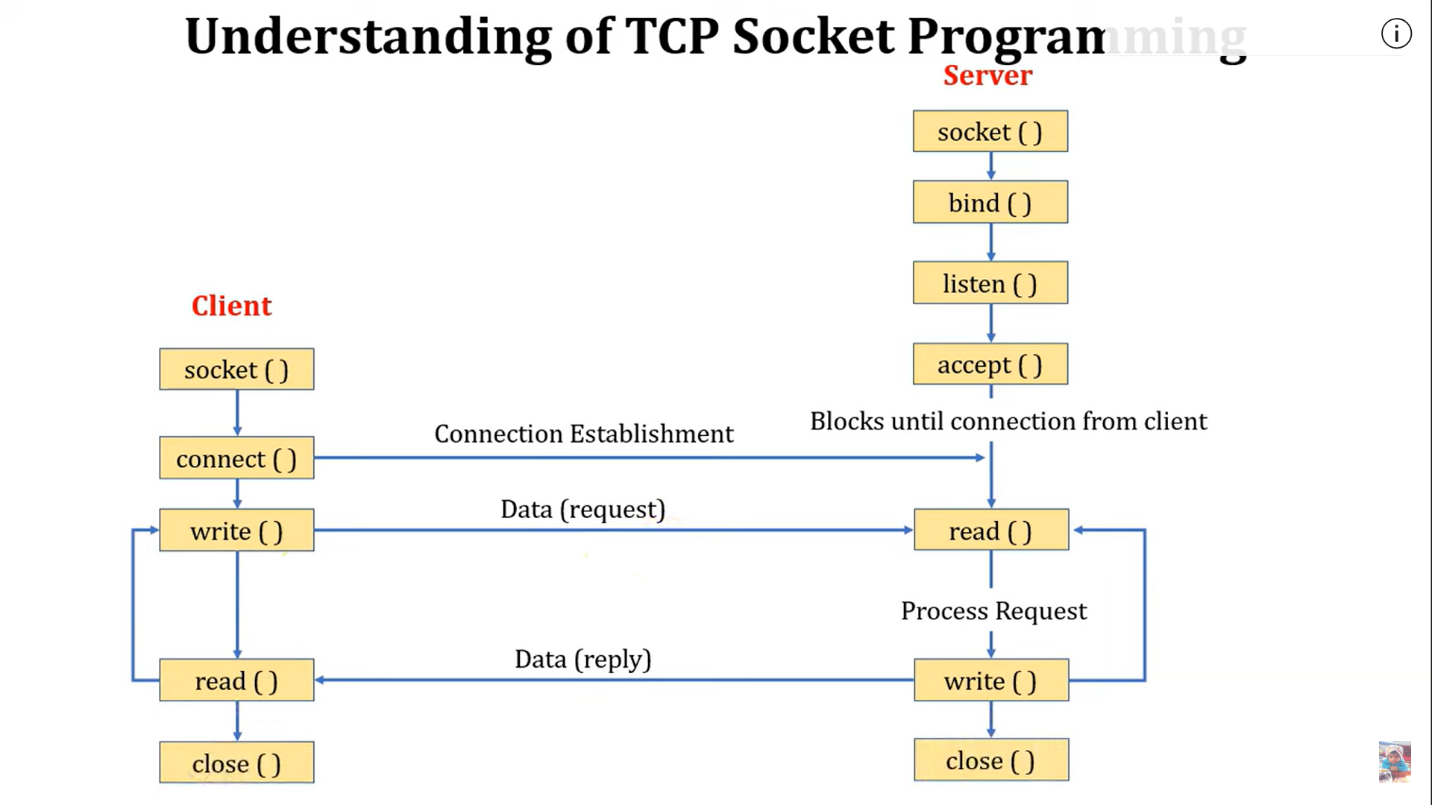
listen() : waiting for the connection

accept(): Accept connection for client

send()/recv()/sendto()/recvfrom() : exchange the data

close(): end of communication

1. Flow of TCP Socket Programing:



Code:

myClient:

import java.io.IOException;

import java.io.\*;

import java.net.Socket;

public class myclient {

    public static void main(String[] args) {

        String serverAddress = "localhost";

        int serverPort = 2000;

        try {

            // Connect to the server

            Socket socket = new Socket(serverAddress, serverPort);

            System.out.println("Connected to server");

          DataOutputStream  d = new  DataOutputStream(socket.getOutputStream());

           d.writeUTF("Hello from client");

            socket.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

myServer:

import java.io.\*;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.\*;

import java.net.Socket;

public class myserver {

    public static void main(String[] args) {

        int portNumber = 2000;

        try {

            ServerSocket serverSocket = new ServerSocket(portNumber);

            System.out.println("Server listening on port " + portNumber);

            // Wait for client connection

            Socket clientSocket = serverSocket.accept();

            System.out.println("Client connected");

            // Read the message from the client

             DataInputStream  d = new  DataInputStream(clientSocket.getInputStream());

           String str = (String)d.readUTF();

           System.out.println("Message from Client:   " +  str);

        clientSocket.close();

            serverSocket.close();

        } catch (IOException e) {

            e.printStackTrace();

        }

    }

}

Output:

