**1.TCP : Find if the entered number is prime or not.**

**ServerSide**

import java.io.\*;

import java.net.\*;

public class TCPPrimeServer {

public static void main(String[] args) throws IOException {

ServerSocket serverSocket = new ServerSocket(5000);

System.out.println("Server is running on port 5000...");

Socket socket = serverSocket.accept();

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

DataInputStream dis = new DataInputStream(socket.getInputStream());

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

int num = dis.readInt();

int flag = 1;

if (num <= 1) {

flag = 0;

} else {

for (int i = 2; i <= num / 2; i++) {

if (num % i == 0) {

flag = 0;

break;

}

}

}

if (flag == 1)

dos.writeUTF(num + " is a Prime number.");

else

dos.writeUTF(num + " is NOT a Prime number.");

dis.close();

dos.close();

socket.close();

serverSocket.close();

}

}

**ClientSide**

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class TCPPrimeClient {

public static void main(String[] args) throws IOException {

Socket socket = new Socket("localhost", 5000);

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

DataInputStream dis = new DataInputStream(socket.getInputStream());

Scanner sc = new Scanner(System.in);

System.out.print("Enter a number to check prime: ");

int num = sc.nextInt();

dos.writeInt(num);

String result = dis.readUTF();

System.out.println("Result from server: " + result);

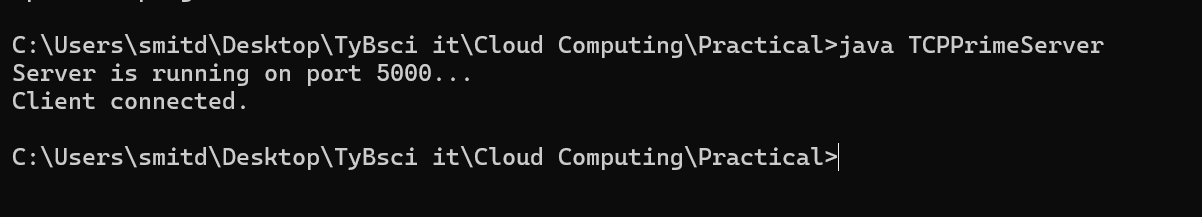
dis.close();

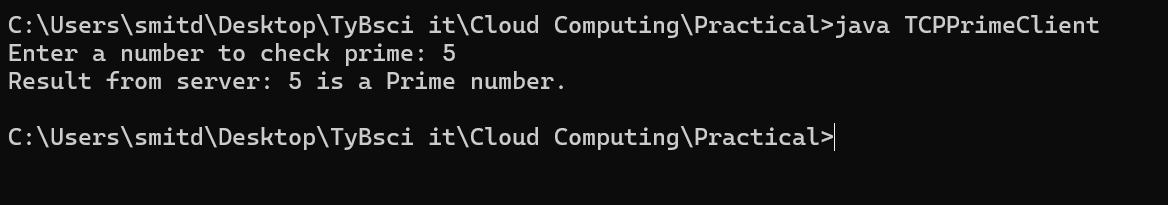
dos.close();

socket.close();

}

}





**2.TCP : A chatting application**

**ServerSide**

import java.io.\*;

import java.net.\*;

public class TCPChatServer {

public static void main(String[] args) throws IOException {

ServerSocket ss = new ServerSocket(6000);

System.out.println("Server waiting for client...");

Socket socket = ss.accept();

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

BufferedReader inputClient = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter outputClient = new PrintWriter(socket.getOutputStream(), true);

BufferedReader inputServer = new BufferedReader(new InputStreamReader(System.in));

String msgFromClient, msgToClient;

while (true) {

msgFromClient = inputClient.readLine();

if (msgFromClient.equalsIgnoreCase("exit")) {

System.out.println("Client left the chat.");

break;

}

System.out.println("Client: " + msgFromClient);

System.out.print("Server: ");

msgToClient = inputServer.readLine();

outputClient.println(msgToClient);

}

socket.close();

ss.close();

}

}

**ClientSide**

import java.io.\*;

import java.net.\*;

public class TCPChatClient {

public static void main(String[] args) throws IOException {

Socket socket = new Socket("localhost", 6000);

BufferedReader inputUser = new BufferedReader(new InputStreamReader(System.in));

PrintWriter outputServer = new PrintWriter(socket.getOutputStream(), true);

BufferedReader inputServer = new BufferedReader(new InputStreamReader(socket.getInputStream()));

String msgToServer, msgFromServer;

while (true) {

System.out.print("Client: ");

msgToServer = inputUser.readLine();

outputServer.println(msgToServer);

if (msgToServer.equalsIgnoreCase("exit")) {

break;

}

msgFromServer = inputServer.readLine();

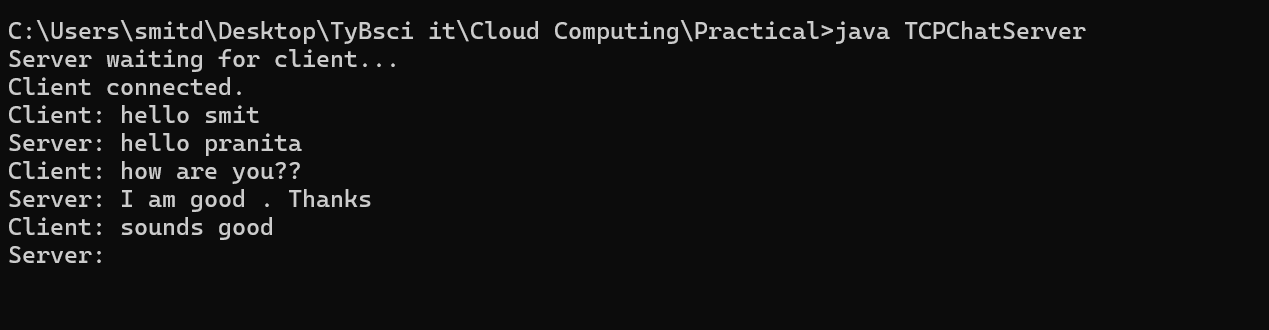
System.out.println("Server: " + msgFromServer);

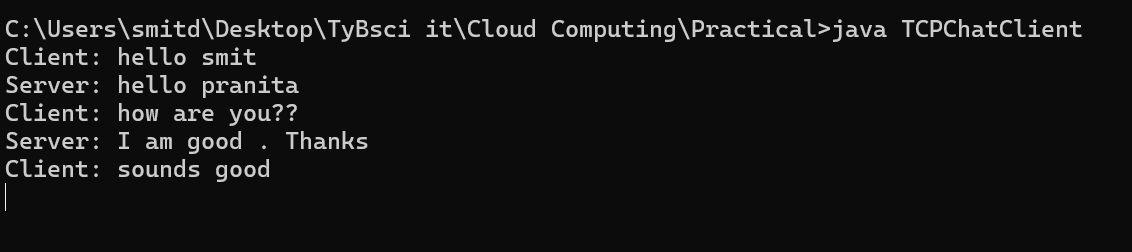
}

socket.close();

}

}





**3. UDP: Find the entered number is ever or odd.**

**ServerSide**

import java.net.\*;

public class UDPEvenServer {

public static void main(String[] args) throws Exception {

DatagramSocket socket = new DatagramSocket(7000);

byte[] buffer = new byte[1024];

System.out.println("UDP Server is running...");

DatagramPacket packet = new DatagramPacket(buffer, buffer.length);

socket.receive(packet);

String received = new String(packet.getData(), 0, packet.getLength());

int num = Integer.parseInt(received);

String response;

if (num % 2 == 0) {

response = num + " is an Even number.";

} else {

response = num + " is an Odd number.";

}

byte[] responseBytes = response.getBytes();

DatagramPacket reply = new DatagramPacket(responseBytes, responseBytes.length, packet.getAddress(), packet.getPort());

socket.send(reply);

socket.close();

}

}

**ClientSide**

import java.net.\*;

import java.util.Scanner;

public class UDPEvenClient {

public static void main(String[] args) throws Exception {

DatagramSocket socket = new DatagramSocket();

Scanner sc = new Scanner(System.in);

System.out.print("Enter a number to check even or odd: ");

int num = sc.nextInt();

String numStr = Integer.toString(num);

InetAddress address = InetAddress.getByName("localhost");

byte[] sendBytes = numStr.getBytes();

DatagramPacket packet = new DatagramPacket(sendBytes, sendBytes.length, address, 7000);

socket.send(packet);

byte[] buffer = new byte[1024];

DatagramPacket response = new DatagramPacket(buffer, buffer.length);

socket.receive(response);

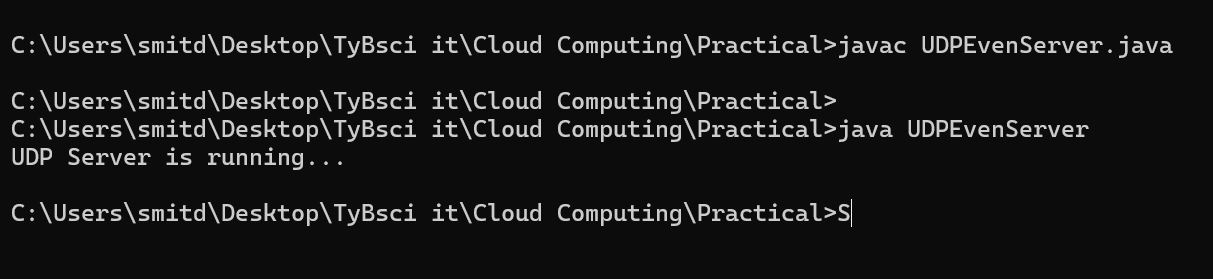
String result = new String(response.getData(), 0, response.getLength());

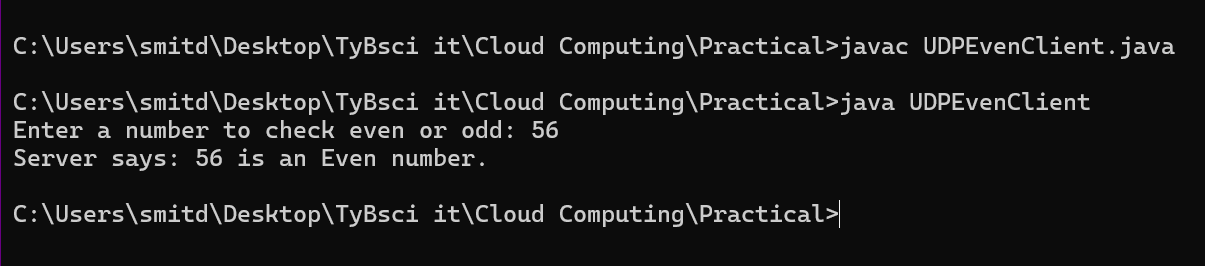
System.out.println("Server says: " + result);

socket.close();

}

}





**4. UDP : Find the factorial number of the given number.**

**ServerSide**

import java.io.\*;

import java.net.\*;

public class FactorialServer {

public static void main(String[] args) throws IOException {

ServerSocket serverSocket = new ServerSocket(5000);

System.out.println("Server started. Waiting for client...");

Socket socket = serverSocket.accept();

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

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

String input = in.readLine();

int number = Integer.parseInt(input);

long fact = 1;

for (int i = 1; i <= number; i++) {

fact \*= i;

}

out.println("Factorial of " + number + " is: " + fact);

socket.close();

serverSocket.close();

}

}

**ClientSide**

import java.io.\*;

import java.net.\*;

public class FactorialClient {

public static void main(String[] args) throws IOException {

Socket socket = new Socket("localhost", 5000);

BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

System.out.print("Enter a number: ");

String number = userInput.readLine();

out.println(number);

String response = in.readLine();

System.out.println(response);

socket.close();

}

}

