



YASHWANTRAO BHONSALE INSTITUTE OF TECHNOLOGY

(DTE CODE: 3470) (MSBTE Code: 1742)

Approved by AICTE, DTE & Affiliated to Mumbai University & MSBTE Mumbai (NBA Accredited ME, CE, EE Diploma Programs)

Experiment No. 8

Aim: Socket programming using TCP:

Resource required: Java Development Kit (JDK), Text Editor / IDE, Command Prompt/ Terminal, Network

Theory:

Java: Java is a programming language and platform used to develop applications that can run on any device with a Java Virtual Machine (JVM).

Socket Programming: Socket Programming is a way to allow two computers (or programs) to communicate with each other over a network.

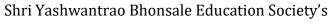
- A socket is like an endpoint a door through which data enters or leaves a program.
- One side acts as a Server, which waits for connections.
- The other side acts as a Client, which connects to the server.

TCP (**Transmission Control Protocol**): TCP (Transmission Control Protocol) is connection-oriented — the client must connect before sending data. We'll create:

- TCPServer.java \rightarrow Waits for connection and replies to client.
- TCPClient.java → Connects to server and sends a message.

TCP Server Code — TCPServer.java

ServerSocket serverSocket = new ServerSocket(6789);



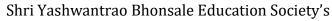


YASHWANTRAO BHONSALE INSTITUTE OF TECHNOLOGY

(DTE CODE: 3470) (MSBTE Code: 1742)

Approved by AICTE, DTE & Affiliated to Mumbai University &MSBTE Mumbai (NBA Accredited ME, CE, EE Diploma Programs)

```
System.out.println("Server is waiting for client connection...");
       Socket socket = serverSocket.accept(); // Wait for client
       System.out.println("Connected to: " + socket.getInetAddress());
       // Input from client
       BufferedReader
                                                                   BufferedReader(new
                                in
                                                      new
InputStreamReader(socket.getInputStream()));
       // Output to client
       PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
       String clientMessage = in.readLine();
       System.out.println("ClientStaff says from BKC: " + clientMessage);
       out.println("Hello Client Staff from BKC, message received!");
       socket.close();
       serverSocket.close();
     } catch (Exception e) {
       System.out.println(e);
}
```





YASHWANTRAO BHONSALE INSTITUTE OF TECHNOLOGY

(DTE CODE: 3470) (MSBTE Code: 1742)

Approved by AICTE, DTE & Affiliated to Mumbai University &MSBTE Mumbai (NBA Accredited ME, CE, EE Diploma Programs)

TCP Client Code — TCPClient.java

```
import java.io.*;
import java.net.*;
public class TCPClient {
  public static void main(String[] args) {
     try {
       Socket socket = new Socket("localhost", 6789);
       // Output to server
       PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
       // Input from server
       BufferedReader
                                                                   BufferedReader(new
                                                      new
InputStreamReader(socket.getInputStream()));
       out.println("Hello Server! This is Client Staff from BKC.");
       String response = in.readLine();
       System.out.println("Server says: " + response);
       socket.close();
     } catch (Exception e) {
       System.out.println(e);
}
```



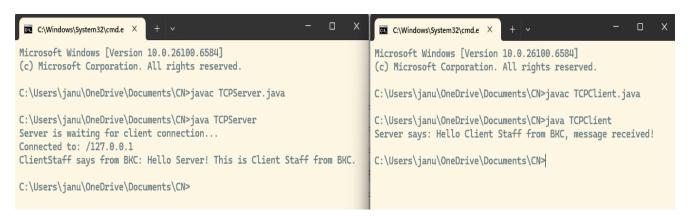
Shri Yashwantrao Bhonsale Education Society's

YASHWANTRAO BHONSALE INSTITUTE OF TECHNOLOGY

(DTE CODE: 3470) (MSBTE Code: 1742)

Approved by AICTE, DTE & Affiliated to Mumbai University &MSBTE Mumbai (NBA Accredited ME, CE, EE Diploma Programs)

Output:



Conclusion:

In this practical, we successfully implemented Socket Programming using TCP (Transmission Control Protocol) to establish communication between a client and a server.

The client sent a message to the server through a TCP connection, and the server received it successfully, proving that data transmission over a reliable, connection-oriented network is possible using sockets.

This experiment helped us understand:

- How to create sockets using Java's ServerSocket and Socket classes.
- How data is transmitted using input and output streams.
- The concept of client-server communication over TCP.

Thus, the practical demonstrated reliable, ordered, and error-free data communication using TCP socket programming in Java.