Sockets

In Java, sockets are a mechanism for communication between two computers or between a computer and a program. Sockets provide a low-level networking API for sending and receiving data over a network. Java supports both TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) sockets.

Here's a brief overview of the concepts related to sockets in Java:

1. **Socket Basics :**
    - A socket is an endpoint for sending or receiving data across a computer network.
    - In Java, the **Socket** class is used for client-side communication, and the **ServerSocket** class is used for server-side communication.
    - Sockets are categorized into two main types: TCP sockets for reliable, connection-oriented communication, and UDP sockets for connectionless communication.
2. **TCP Sockets :**
    - TCP is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data.
    - For TCP communication, a server creates a **ServerSocket** to listen for incoming connections, and a client creates a **Socket** to connect to the server.
    - The server accepts incoming connections using **accept()** method of **ServerSocket**, and communication is performed through the **InputStream** and **OutputStream** of the
    - is done by sending and receiving packets of data using these classes.
3. **Exception Handling:**
    - Socket operations may throw exceptions, so it's important to handle exceptions properly in your code. Common exceptions include **IOException** for general socket errors.
4. **Closing Sockets:**
    - Always close sockets when they are no longer needed. Use the **close()** method to release resources.

Example:

javaCopy code

socket.close(); // Close the socket

These are the basic concepts of sockets in Java. Understanding these concepts allows you to create networked applications for various purposes, such as client-server communication or distributed systems. Keep in mind that proper error handling, resource management, and security considerations are essential when working with sockets.

Run CODES:

Run the TWO code given “Server” first , then “Client”