🧪 EXPERIMENT 5 — TCP Socket Programming

Aim:
Write a program using **TCP socket** for wired network to
a) Say Hello to each other
b) Transfer a file

Objectives:

1. Understand the working of sockets.
2. Study different methods associated with Client and Server sockets.

Theory (Summary):

- **TCP (Transmission Control Protocol)** provides reliable, connection-oriented communication between applications.
- **Client-Server Model:**
  - **Client:** initiates the request.
  - **Server:** waits and responds.
- **Socket:** an endpoint for communication between two machines.

Client-side steps:

1. Create socket → socket()
2. Connect to server → connect()
3. Communicate → read() / write()
4. Close socket

Server-side steps:

1. Create socket → socket()
2. Bind socket → bind()
3. Listen for client → listen()
4. Accept connection → accept()
5. Communicate
6. Close socket

Socket Types:

1. **Stream Socket (TCP)** — reliable, ordered delivery.
2. **Datagram Socket (UDP)** — connectionless, fast, unreliable.
3. **Raw Socket** — used for developing new protocols.

Important Methods:

- **Server Socket:** bind(), listen(), accept()
- **Client Socket:** connect()
- **General:** send(), recv(), sendto(), recvfrom(), close()

Conclusion:
TCP socket program implemented successfully to exchange data and say hello.

🔹 Oral / Viva Questions and Answers

1. **What is a socket?**
   ➤ A socket is an endpoint for communication between two programs over a network.
2. **Types of sockets?**
   ➤ Stream (TCP), Datagram (UDP), Raw, and Sequenced Packet sockets.
3. **Difference between connection-oriented and connectionless communication?**
   ➤ TCP is connection-oriented (reliable), UDP is connectionless (faster, unreliable).
4. **Why is a port number required?**
   ➤ It identifies specific processes or services on a host.
5. **Methods associated with Server Socket?**
   ➤ bind(), listen(), accept().
6. **Methods associated with Client Socket?**
   ➤ connect(), send(), recv(), close().
7. **Which layer does TCP operate on?**
   ➤ Transport layer of the OSI model.