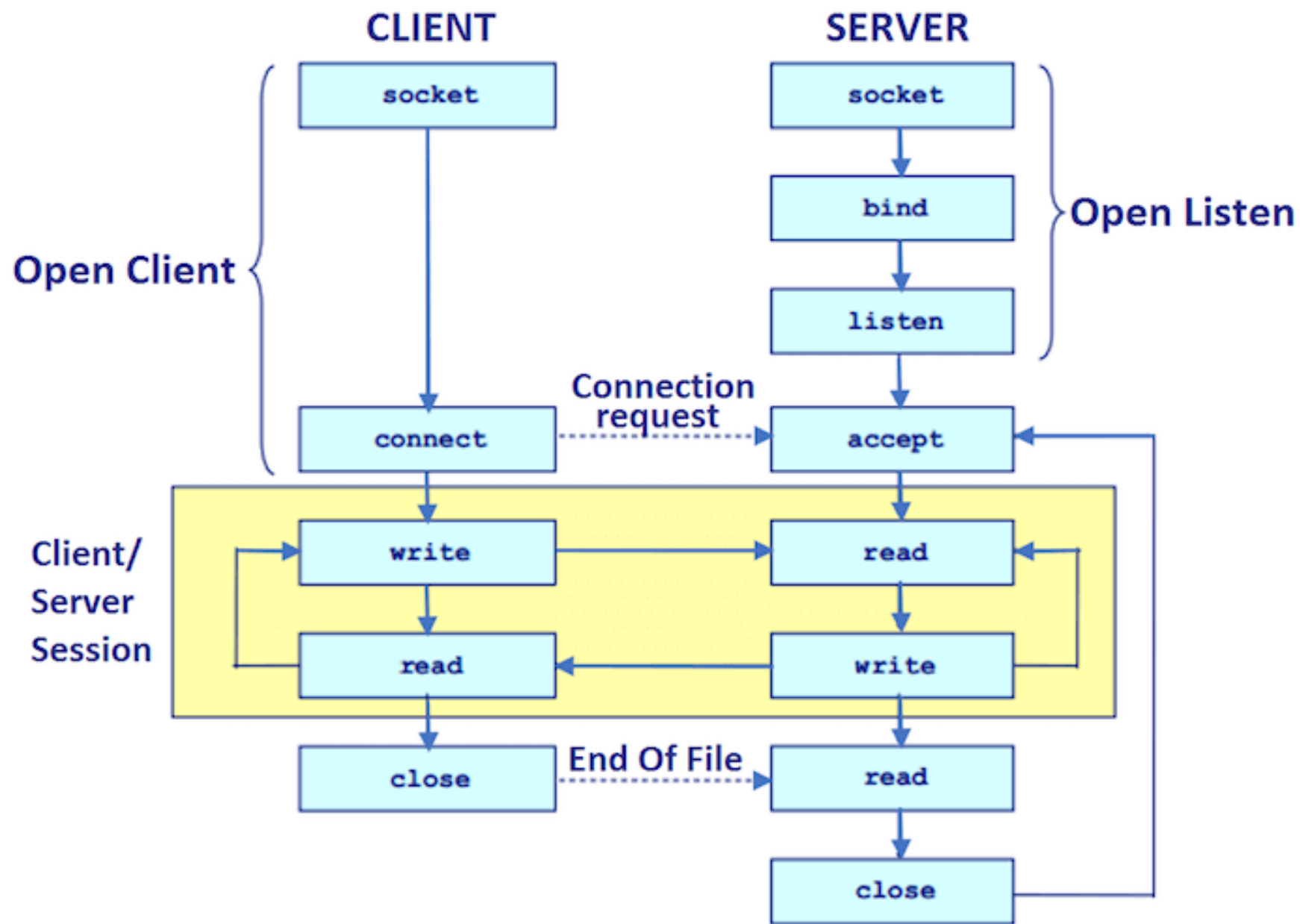# Socket Programming

- A <mark>socket</mark> is a communications connection point (endpoint) that you can name and address in a network.

-

- Socket programming shows <mark>how to use socket APIs to establish communication links</mark> between remote and local processes.

- <mark>Socket programming can be connection-oriented or connection-less</mark>.

- The processes that <mark>use a socket can reside on the same system or different systems</mark> on different networks.

Connection Oriented Service

SOCKET API

Using Socket Programming implement the connection-oriented service using standard ports in any programming language (C,C++, JAVA,Python etc).

Using Java

Example 1

Server Side

```java
import java.net.*;
import java.io.*;
class MyServer{
public static void main(String args[])throws Exception{
ServerSocket ss=new ServerSocket(3333); //Creates a server socket, bound to the specified port
Socket s=ss.accept(); //Listens for a connection to be made to this socket and accepts it
DataInputStream din=new DataInputStream(s.getInputStream()); //Creates a Data Input Stream
DataOutputStream dout=new DataOutputStream(s.getOutputStream()); //Creates a new data output stream to
write data
BufferedReader br=new BufferedReader(new InputStreamReader(System.in)); //Creates a buffering
character-input stream
String str="",str2=""; //two string type variable declarations
while(!str.equals("stop")){  //until the input by the client is not stop
str=din.readUTF(); //Writes a string to the underlying output stream using modified UTF-8 encoding
System.out.println("client says: "+str);
str2=br.readLine(); //Reads a line of text
dout.writeUTF(str2); //Writes a string to the underlying output stream using modified UTF-8encoding
dout.flush(); //Flushes this data output stream
}
din.close();
s.close();
ss.close();
}}
```

Client Side

```java
import java.net.*;
import java.io.*;
class MyClient{
public static void main(String args[])throws Exception{
Socket s=new Socket("localhost",3333); //Creates a stream socket and connects it to the
specified portnumber on the named host
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("stop")){
str=br.readLine();
dout.writeUTF(str);
dout.flush();
str2=din.readUTF();
System.out.println("Server says: "+str2);
}

dout.close();
s.close();
}}
```

```java
import java.net.*;
class MyClient{
public static void main(String args[])throws Exception{
Socket s=new Socket("localhost",222);
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("stop")){
str=br.readLine();
dout.writeUTF(str);
dout.flush();
```

Console ×

MyServer [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:36:34 pm) [pid: 9100]

Markers   Properties   Servers   Snippets   Terminal   Console ×

MyClient [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:36:45 pm) [pid: 32792]

Hey

```java
import java.net.*;
class MyClient{
public static void main(String args[])throws Exception{
Socket s=new Socket("localhost",222);
DataInputStream din=new DataInputStream(s.getInputStream());
DataOutputStream dout=new DataOutputStream(s.getOutputStream());
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

String str="",str2="";
while(!str.equals("stop")){
str=br.readLine();
dout.writeUTF(str);
dout.flush();
```

◄

⊟ Console ✕

MyServer [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:36:34 pm) [pid: 9100]

```
client says: Hey
```

◄

⧉ Markers ▣ Properties ⚙ Servers ⬚ Snippets ⌨ Terminal ⊟ Console ✕

MyClient [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:36:45 pm) [pid: 32792]

```
Hey
```

Example 2

Server Side

```java
// A Java program for a Server
import java.net.*;
import java.io.*;
public class Server
{
//initialize socket and input stream
private Socket socket = null;
private ServerSocket server = null;
private DataInputStream in = null;
// constructor with port
public Server(int port)
{
// starts server and waits for a connection
try{
server = new ServerSocket(port);
System.out.println("Server started");
System.out.println("Waiting for a client ...");
socket = server.accept();
System.out.println("Client accepted");
// takes input from the client socket
in = new DataInputStream(
new BufferedInputStream(socket.getInputStream()));
String line = "";
// reads message from client until "Over" is sent
while (!line.equals("Over")){
try{
line = in.readUTF();
System.out.println(line);
}
catch(IOException i){
System.out.println(i);
} }
System.out.println("Closing connection");
// close connection
socket.close();
in.close();
}
```

```java
catch(IOException i)
{
System.out.println(i);
}
}
public static void main(String args[])
{
Server server = new Server(5000);
}
}
```

Client Side

```java
// A Java program for a Client
import java.net.*;
import java.io.*;
public class Client
{
// initialize socket and input output streams
private Socket socket = null;
private BufferedReader input = null;
private DataOutputStream out = null;
// constructor to put ip address and port
public Client(String address, int port)

{
// establish a connection
try
{
socket = new Socket(address, port);
System.out.println("Connected");

// takes input from terminal
input = new BufferedReader(new
InputStreamReader(System.in));;
// sends output to the socket
out = new
DataOutputStream(socket.getOutputStream());
}
catch(UnknownHostException u){
System.out.println(u);
}
catch(IOException i){
System.out.println(i);
}
// string to read message from input
String line = "";
```

```java
// keep reading until "Over" is input
while (!line.equals("Over"))
{
try
{
line = input.readLine();
out.writeUTF(line);
}
catch(IOException i)
{
System.out.println(i);
}
}

// close the connection
try
{
input.close();
out.close();
socket.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
public static void main(String args[])
{
Client client = new Client("127.0.0.1", 5000);
}
}
```

Client.java   Server.java ×

```java
32  line = in.readUTF();
33  System.out.println(line);
34  }
35  catch(IOException i)
36
37  {
38  System.out.println(i);
39  }
```

Console ×

Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.ex

```
Server started
Waiting for a client ...
```

Markers   Properties   Servers   Snippets   Terminal   Cor

Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.ex

```
Server started
Waiting for a client ...
```

Client.java × **Server.java**

```java
43  }
44  catch(IOException i)
45  {
46  System.out.println(i);
47  }
48  }
49  // close the connection
50
```

Console ×

Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe

```
Server started
Waiting for a client ...
Client accepted
```

Markers   Properties   Servers   Snippets   Terminal   Conso

Client [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe

```
Connected
```

Client.java ✕   Server.java

```
43 }
44 catch(IOException i)
45 {
46 System.out.println(i);
47 }
48 }
49 // close the connection
50 try
```

◀

🖥 Console ✕

Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\

```
Server started
Waiting for a client ...
Client accepted
```

◀

🔖 Markers ▦ Properties ⚙ Servers 🖹 Snippets 💻 Term

Client [Java Application] C:\Program Files\Java\jdk-18.0.1.1\

```
Connected
Hey
```

Client.java ✕ **Server.java**

```
43  }
44  catch(IOException i)
45  {
46  System.out.println(i);
47  }
48  }
49  // close the connection
```

Console ✕

Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.ex

```
Server started
Waiting for a client ...
Client accepted
Hey
```

Markers **Properties** **Servers** **Snippets** **Terminal** **Con**

Client [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.ex

```
Connected
Hey
```

```
43 }
44 catch(IOException i)
45 {
46 System.out.println(i);
47 }
48 }
49 // close the connection
```

Console ×

Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-No

```
Server started
Waiting for a client ...
Client accepted
Hey
How are you ?
```

Markers Properties Servers Snippets Terminal Console ×

Client [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-No

```
Connected
Hey
How are you ?
```

```
48 }
49 // close the connection
50 try
51 {
52 input.close();
53 out.close();
54 socket.close();
```

◀

Console ✕

<terminated> Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-I

```
Server started
Waiting for a client ...
Client accepted
Hey
How are you ?
Over
Closing connection
```

◀

Markers   Properties   Servers   Snippets   Terminal   Console ✕

<terminated> Client [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-N

```
Connected
Hey
How are you ?
Over
```

Connection-less Service

Using Socket Programming implement the connectionless service using standard ports in any programming language (C,C++, JAVA,Python etc).

Using Java

Sender Side

```java
import java.net.*;
import java.util.Scanner;
public class DSender{
  public static void main(String[] args) throws Exception {
      Scanner sc = new Scanner(System.in);

      //Constructs a datagram socket and binds it to any available portion the local
      host machine
      DatagramSocket ds = new DatagramSocket();
      String str = sc.next();
      //Determines the IP address of a host, given the host's name/ or determines the
      validity of the address format, given the literal IP address
      InetAddress ip = InetAddress.getByName("127.0.0.1");
      //Constructs a datagram packet for sending packets of length to the specified
      port number on the specified host
      DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(), ip, 123);
      // Sends a datagram packet from this socket
      ds.send(dp);
      ds.close();
  } }
```

Receiver Side

```java
import java.net.*;
public class DReceiver{
  public static void main(String[] args) throws Exception {
    //Constructs a datagram socket and binds it to the specified port on the local host
    machine
    DatagramSocket ds = new DatagramSocket(123);
    //Generating a buffer of size 1024 bytes
    byte[] buf = new byte[1024];
    //Constructs a DatagramPacket for receiving packets of length length
    DatagramPacket dp = new DatagramPacket(buf, 1024);
    //Receives a datagram packet from this socket
    ds.receive(dp);
    String str = new String(dp.getData(), 0, dp.getLength());
    System.out.println("Sender says "+str);
    ds.close();
  }
}
```

```java
import java.net.*;

public class ServerUDP{
  public static void main(String[] args) throws Exception {
    Scanner sc = new Scanner(System.in);

    //Constructs a datagram socket and binds it to the specified port.
    DatagramSocket ds = new DatagramSocket(3000);
    byte[] buf = new byte[1024];
```

Console ×

ServerUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:07:59 pm) [pid: 4924]

Markers Properties Servers Snippets Terminal Console ×

ServerUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:07:59 pm) [pid: 4924]

```
1⊕ import java.net.*;
3  public class ClientUDP{
4⊖   public static void main(String[] args) throws Exception {
5       Scanner sc = new Scanner(System.in);
6       DatagramSocket ds = new DatagramSocket();
7       String str = sc.next();
8       InetAddress ip = InetAddress.getByName("127.0.0.1");
```

◀

📋 Console ✕

ServerUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:07:59 pm) [pid: 4924]

◀

🔲 Markers   🔲 Properties   ⚙ Servers   📄 Snippets   🖳 Terminal   📋 Console ✕

ClientUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:08:38 pm) [pid: 39424]
Hey

```java
1 import java.net.*;
3 public class ClientUDP{
4   public static void main(String[] args) throws Exception {
5     Scanner sc = new Scanner(System.in);
6     DatagramSocket ds = new DatagramSocket();
7     String str = sc.next();
8     InetAddress ip = InetAddress.getByName("127.0.0.1");
```

Console ✕

ServerUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:07:59 pm) [pid: 4924]

Client says Hey

Markers Properties Servers Snippets Terminal Console ✕

<terminated> ClientUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:08:38 pm – 1

Hey

```java
1 import java.net.*;
3 public class ClientUDP{
4   public static void main(String[] args) throws Exception {
5     Scanner sc = new Scanner(System.in);
6     DatagramSocket ds = new DatagramSocket();
7     String str = sc.next();
8     InetAddress ip = InetAddress.getByName("127.0.0.1");
```

Console ✕

ServerUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:07:59 pm) [pid: 4924]

```
Client says Hey
```

Markers   Properties   Servers   Snippets   Terminal   **Console** ✕

<terminated> ClientUDP [Java Application] C:\Progra

```
Hey
```

1 MyServer [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:35:18 pm) [pid: 30948]

2 MyServer [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:36:34 pm) [pid: 9100]

3 MyClient [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:36:45 pm) [pid: 32792]

4 MyClient [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:38:34 pm) [pid: 24176]

5 Server [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:39:43 pm) [pid: 12980]

6 Client [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 10:40:11 pm) [pid: 18152]

7 ServerUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:07:59 pm) [pid: 4924]

8 <terminated> ClientUDP [Java Application] C:\Program Files\Java\jdk-18.0.1.1\bin\javaw.exe (06-Nov-2022, 11:08:38 pm – 11:09:00 pm) [pid: 39424]

Thank You !