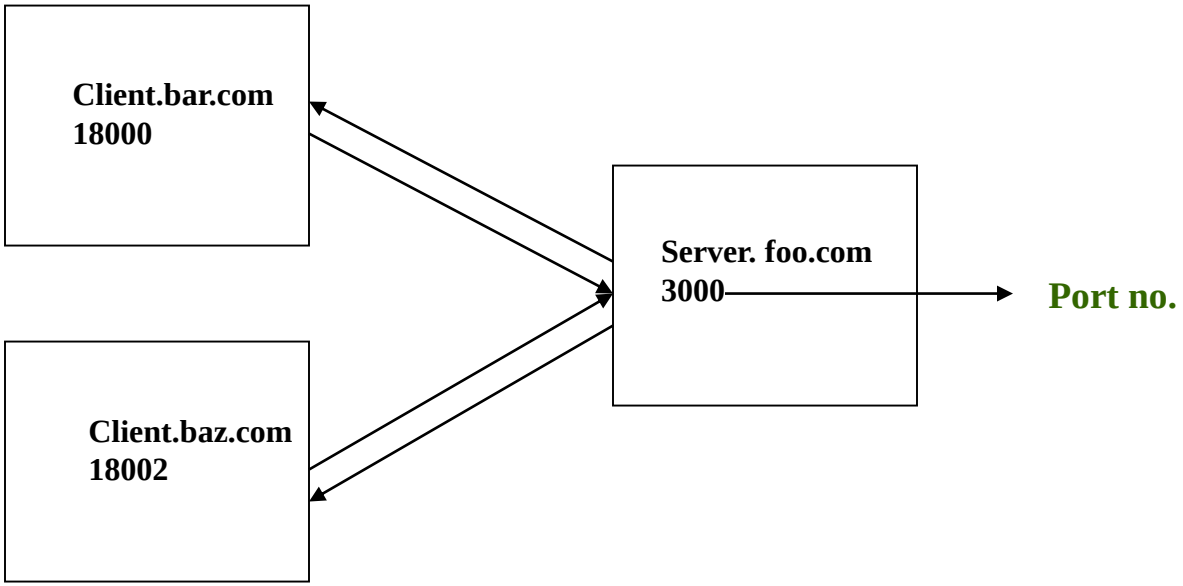
Objectives

- In this session, you will learn to:
  - Describe Input/Output fundamentals
  - Construct node and processing streams, and use them appropriately
  - Distinguish readers and writers from streams, and select appropriately between them
  - Develop code to set up the network connection
  - Understand the TCP/IP Protocol
  - Use ServerSocket and Socket classes for implementation of TCP/IP clients and servers

Networking

- Basics of Networking:
  - Computers running on the Internet communicating to each other using the Transmission Control Protocol (TCP) / Internet Protocol (IP).



Networking (Contd.)

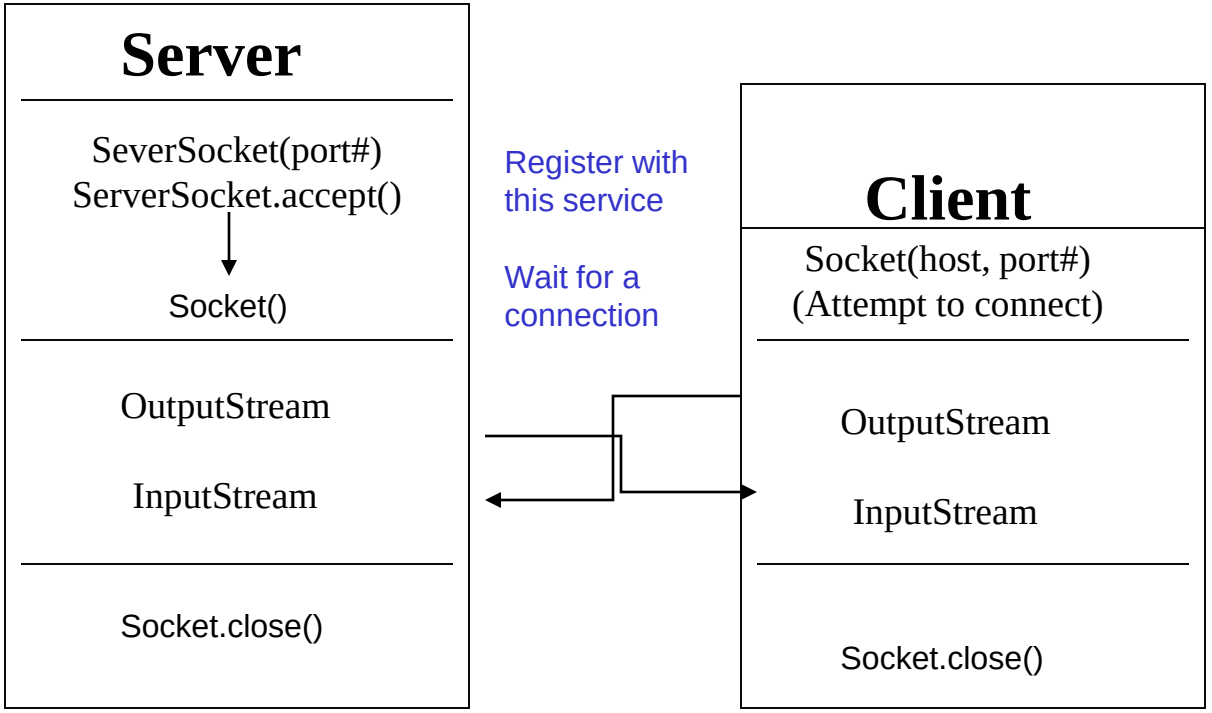
- Networking with Java Technology:
  - Sockets:
    - Sockets hold two streams, an input stream and an output stream.
    - Each end of the socket has a pair of streams.
  - Setting Up the Connection:
    - Setup of a network connection is similar to a telephone.
    - One end must dial the other end, which must be listening.

Networking (Contd.)

- To address the connection, include the following:
  - The address or name of remote machine.
  - A port number to identify the purpose at the server.
- Port numbers range from 0–65535

Networking (Contd.)

- Java Networking Model:



ServerSocket and Socket Classes

- Code Snippet for Creating Minimal TCP/IP Server:

```
ServerSocket s = null;
s = new ServerSocket(5432); //Register your service on port 5432
while (true) // Run the listen/accept loop forever
{
  Socket s1 = s.accept(); // Wait here and listen for a connection
  OutputStream s1out = s1.getOutputStream();
  // Get output stream associated with the socket
  BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(s1out));
  bw.write("Hello Net World!\n"); // Send your string!
```

ServerSocket and Socket Classes (Contd.)

```
    bw.close();  // Close the connection, but not the server
    socket
    s1.close();
}
```

- Code Snippet for Creating Minimal TCP/IP Client:

```
// Open your connection to a server, at port 5432
// localhost used here
Socket s1 = new Socket("127.0.0.1", 5432);
// Get an input stream from the socket
InputStream is = s1.getInputStream();
//Decorate it with a "data" input stream
DataInputStream dis = new
DataInputStream(is);
```

ServerSocket and Socket Classes (Contd.)

```
// Read the input and print it to the screen
System.out.println(dis.readUTF());
// When done, just close the steam and connection
dis.close();
s1.close();
```

Demonstration

Lets see how to create a TCP based Java client, and use user-provided system properties to drive a Java program.