Class Name	Method Name	Explanation
ServerSocket	ServerSocket(int port)	Constructor: Creates a server socket that listens on the specified port.
	ServerSocket(int port, int backlog)	Constructor: Creates with port and specifies the maximum length of the queue for incoming connections (backlog).
	ServerSocket(int port, int backlog, InetAddress bindAddress)	Constructor: Creates with port, backlog, and binds to a specific IP address; if bindAddress is null, binds to all interfaces.
	accept()	Blocking Method: Listens for and accepts a client connection attempt. Returns a new Socket object for the client.
	bind(SocketAddress endpoint)	Binds the server socket to a specific SocketAddress (IP address and port).
	bind(SocketAddress endpoint, int backlog)	Binds to a specific SocketAddress and sets the connection backlog.
	close()	Closes the server socket, releasing resources and stopping it from accepting new connections.
	getInetAddress()	Returns the InetAddress to which the server socket is bound, or null if not bound yet.
	getLocalPort()	Returns the port number on which the server socket is listening.
Socket	Socket(String host, int port)	Constructor: Creates a socket and immediately attempts to connect to the specified host and port.
	Socket(InetAddress address, int port)	Constructor: Creates and connects to a server at the specified InetAddress and port.
	connect(SocketAddress endpoint)	Connects the socket to a server at the given SocketAddress with a default connection timeout.
	close()	Closes the socket, terminating the connection and releasing associated resources.
	getInputStream()	Returns an InputStream to read raw bytes received from the connected server. This is the lowest level input, dealing directly with byte streams.
	getOutputStream()	Returns an OutputStream to send raw bytes to the connected server. This is the lowest level output, dealing directly with byte streams.
	getInetAddress()	Returns the InetAddress of the remote server to which the socket is connected.
	getLocalAddress()	Returns the InetAddress of the local end of the socket.
	getPort()	Returns the port number of the remote server to which the socket is connected.
PrintWriter	PrintWriter(OutputStream out)	Constructor: Creates a PrintWriter that writes formatted characters to the provided byte-based OutputStream. Wraps OutputStreamWriter (implicitly) to handle character encoding (often UTF-8 default, but encoding can be specified via OutputStreamWriter).
	PrintWriter(OutputStream out, boolean autoFlush)	Constructor: Creates with an OutputStream and option for automatic flushing. If autoFlush is true, buffer is flushed after every println().
	println(String s)	Prints a string of characters to the output stream followed by a newline character. If autoFlush is enabled, the buffer is flushed immediately.
	println()	Prints only a newline character to the output stream. If autoFlush is enabled, the buffer is flushed immediately.
	flush()	Manually forces the output buffer to be written to the underlying stream. Essential when autoFlush is off or for operations other than println().
	close()	Closes PrintWriter and underlying output stream. Releases resources.
BufferedReader	BufferedReader(Reader in)	Constructor: Creates a BufferedReader to efficiently read characters. It wraps a Reader. Common Layer 2: Character Stream.
	(constructor parameter Reader in is often)	new InputStreamReader(socket.getInputStream(), "UTF-8") - Layer 1: Byte-to-Character Conversion. InputStreamReader wraps the InputStream from the socket and decodes bytes to characters using a specified character encoding (like "UTF-8"). Crucial for handling text data correctly over sockets.
	(bottommost layer is)	socket.getInputStream() - Layer 0: Byte Stream from Socket. Provides raw bytes received from the network socket connection.
	readLine()	Blocking Method: Reads a line of text. Returns String (line without newline) or null (end of stream). Efficiently reads lines due to buffering. Blocks until a full line or end-of-stream.
	close()	Closes BufferedReader and all underlying layers: Reader (e.g., InputStreamReader) and InputStream (from Socket). Releases all associated resources in the chain.